SYNTHESIS OF A GUST ALLEVIATION SYSTEM USING ACTIVE CONTROLS

by

ALOK KHARE

DEPARTMENT OF AEROSPACE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL 1990

Synthesis Of A Gust Alleviation System Using Active Controls

A thesis submitted
in partial fullfilment of the requirements
for the degree of
Master of Technology

by

Alok Khare

to the

Department of Aerospace Engineering
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

April, 1990

Certificate

This is to certify that this work entitled ***Synthesis of a Gust Alleviation System Using Active Controls*** is the record of the work done by Alok Khare (*Roll No. 8810101*) under my supervision and it has not been submitted elsewhere for the award of a degree.

April 1990

(C V R Murti)

Assistant Professor

Department of Aerospace Engineering

Indian Institute of Technology

Kanpur 208016

INDIA

Acknowledgements

I wish to thank Mr. C V R Murti for his invaluable guidance and encouragement throughout this work.

I wish to thank Mr. Narendra Gupta for patiently proof reading the writeup.

My sincere thanks to all my friends for their help and companionship during my stay at I.I.T. Kanpur.

—ALOK KHARE

Contents

Appendices

List of Figures

List of Tables

List of Symbols

$\bar{c}$	mean aerodynamic chord	ft
g	acceleration due to gravity	ft sec^{-2}
h	altitude	ft
m	mass of the aircraft	slugs
i_B	non dimensional moment of inertia	
n	loadfactor	
S	Wing planform area	ft^2
U_e	uniform forward velocity	ft sec^{-1}
$\hat{u}$ $\hat{w}$ $\hat{q}$	non dimensional quantities	
δ_e (or η)	elevator deflection	rad
δ_{flap}	flap deflection	rad
t^*	characteristic time	
l	characteristic length	
μ	relative mass parameter	

NACA stability derivatives have been used throughout this work. The non dimensionalisation of the stability derivatives is defined in Appendix B.

Abstract

The project aims at synthesis of a control law for gust alleviation of aircraft. The currently favoured Von Karman model of the gust is being used for mathematical modelling of the gust. . The gust is assumed to be uniform in the spanwise direction and varies only along the vertical direction ie. a vertical gust. The statistical properties of gust are described using the power spectral approach. The response of the aircraft to the gust thus modelled is studied in terms of motions, accelerations, rate of change of accelerations and other dynamical characteristics such as passenger comfort. A rigid six degree of freedom model of the aircraft is being used whose longitudinal dynamics are being considered at present. The performance of the gust alleviation system is evaluated by finding the response of the aircraft when it encounters a gust of known spectral density specified by the airworthiness requirements. The gust would be sensed by sensors such as vanes or alternatively, the aircraft motions sensed by accelerometers and gyros and these will be fed as inputs to the gust alleviaton system. Either of the methods may be used or their combination. Deflection of control surfaces such as elevator, flaps, etc. is being used to counteract the lift increments due to the gusts and thus to help in gust alleviation. The deflection of controls is governed by the proposed control law (which is both feedforward and feedback).An exploratory study of the effect of relaxed static stability on gust alleviation was also made but the results show that superaugumentation does not help in gust alleviation.

The present project emphasises on the reduction of normal accelerations to improve passenger comfort rather than on the reduction of structural stresses.

The criterion for passenger comfort proposed by Krag is adopted. A combination of the feedforward and feedback control law is proposed for the gust alleviation system with the objective to improve passenger comfort. This does not deteriorate the handling qualities. The feedforward portion senses the gust angle of attack and feedforward control is applied through direct lift control devices such as flaps while the feedback portion senses the angle of attack and feeds it back to the elevator. The best combination of the appropriate gains of feedback and feedforward controls is obtained towards minimization of a passenger comfort criterion. The determination of the optimal gain values is made by the application of DownHill Simplex Method; using the passenger comfort criterion proposed by Krag as the objective function. The results show that the combination of feedforward loop of gust angle attack to the flaps and a feedback of the angle of attack to the elevator reduces the passenger comfort index from 2.17146 for the gust alleviation system disengaged to 1.63278 for the gust alleviation system engaged which is quite a significant reduction without much affecting the handling qualities.

Chapter 1. Introduction

The present work is an attempt for synthesis of a control law for gust alleviation of aircraft and to explore the effect of relaxed static stability on gust alleviation.

1.1 Introduction

The airmass through which airplanes fly is constantly in a state of flux. Its motion is variable both in time and space, random and exceedingly complex. This motion is called turbulence. The gradients associated with this atmospheric turbulence are termed gusts. When an airplane encounters a gust, large sudden loads arise on it which may cause failure of the airframe due to accumulation of fatigue, deterioration of the handling and riding qualities and especially the controllability of the airplane in strong turbulent wind.

1.2 Literature Survey

The problem of aircraft response to gusts is one of the the oldest because of its strong influence on airplane strength, structural fatigue life, pilot & passenger comfort, navigation and aiming, and weapon delivery. The problem was recognized as a high priority item by NACA when it was originally established.[1] Gust alleviation is of considerable interest for small transport aircraft which fly at medium altitudes where most turbulence is encountered. For these aircraft, the improvement of passenger ride comfort is the main aim of gust alleviation which requires reduction of sustained changes in vertical acceleration.[2]

Because of widespread effects of gust encounter, there has always been a

continuing interest for devising means to alleviate or control the loads and the motions that result. Essentially, there are two types of gust response control .

(a) Sensors such as vanes, laser doppler anemometer etc. may be used to sense the gusts, its signal being used to control some auxiliary lift device which nullifies the forces on the airplane due to the gusts — a typical feedforward process

(b) some aircraft motion may be sensed by accelerometers and gyros, which in turn is used to activate some control which counteracts the motion - a typical feedback loop process

The two forms may be used separately or in combination. The direct lift controls (DLC) such as flaps, spoilers, etc. are more of the gust alleviation type. The force applicators (eg. control surfaces) provide either motion control (such as the familiar yaw dampers) or may be used to supress modal response. Geometric change is limited to very few configurations.[3] The structure of the gust response problem is conveniently subdivided as is illustrated in Fig 1.1.[4]

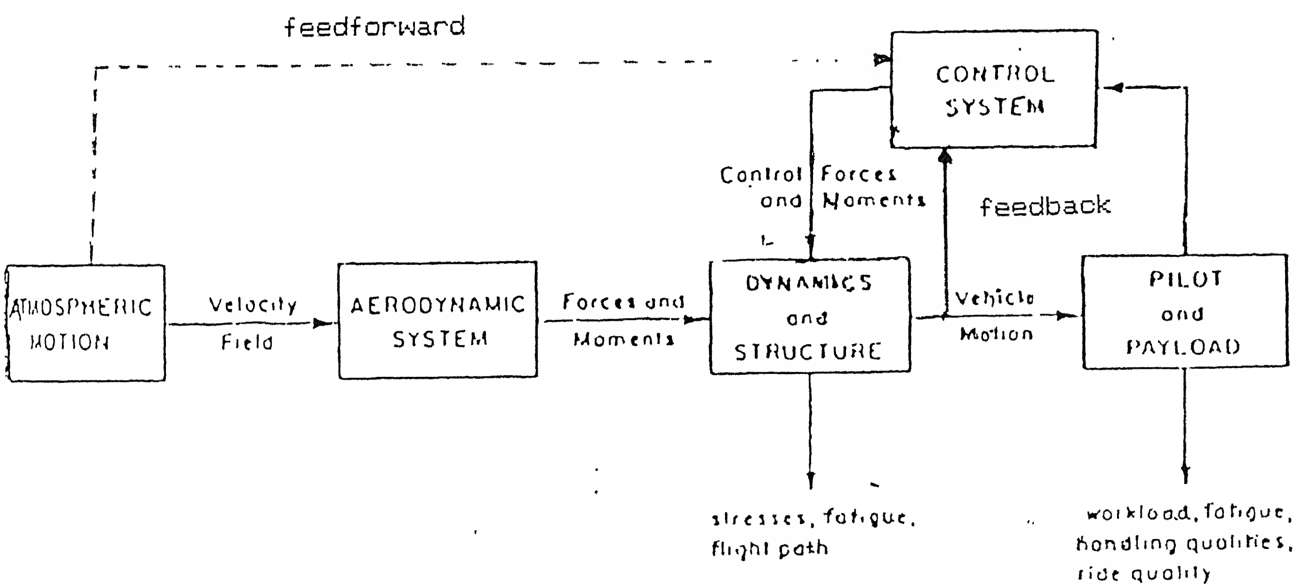


Fig 1.1. Structure of the gust response problem

Gust compensation is the most efficient method of gust alleviation. A compensation

of the gust induced lift can be accomplished by a proper deflection of the control surface at the very moment the gust begins to act on the wing . At the same time, the control surface creates the same lift as the gust but acting in the opposite direction.[2] Preliminary investigations by Krag[2] have shown that for small transport aircraft, a mixed open loop and closed loop system might be promising. The open loop portion performs a gross alleviation while the closed loop portion is responsible for the suppression of residual motions. The gust alleviation system affects mainly the short period motion of the aircraft. The vertical acceleration is fed back to the control surfaces. Wing control surfaces and the elevator are acting in DLC mode. The pitch acceleration generated by the gust is not alleviated by this means. This could be accomplished by pitch acceleration feedback to the elevator. Both feedbacks are disadvantageous with respect to the handling qualities and controllability of the aircraft. This is the reason why $\ddot{\theta}$ feed back is not considered. Because of its static stability, the aircraft tends to turn into the direction of the airflow. This behaviour auguments the gust alleviation and therefore should not be suppressed by a $\ddot{\theta}$ feedback. An additional small $\dot{\theta}$ feedback would improve the situation without the handling qualities further deteriorating. The $\ddot{z}$ feedback has a de-stabilizing effect on the phugoid motion and decreases the natural frequency of the short period motion.[6] According to B. Krag, the dynamic response characteristic of the aircraft can be improved by feeding back θ and $\dot{\theta}$ to the elevator.

The following methods are also proposed to reduce the vertical accelerations due to rough air[3] :

(a) Pitching the whole airplane to maintain a constant angle of attack during passage through the gusts. This is advantageous in the sense that it may be accomplsihed by the use of the elevator without provision of additional controls on the airplane.

(b) Variation of wing incidence to maintain a constant angle of attack during passage through the gusts.

(c) Operation of flaps or other controls to offset the lift increments on the wing.

A problem of longitudinal control arises in connection with the methods (b) & (c).

A mechanism sensitive to changes in acceleration should be most effective for reducing these changes because it could counteract accelerations from any type of gust.

The concept of a power spectrum for the compact representation of a random disturbance in time has been successfully used to find gust loads on airplanes[5]. John Tukey has developed a new technique for the derivation of spectra estimates from the time history data in which an initial raw estimate of the power spectral density is obtained using the Fourier cosine series and this estimate is further smoothened by a more effective sharper filter to reduce the aliasing problem encountered while finding the power spectrum of the signals in the time domain. The author of the same paper also decribes the effect of C.G. position, short period damping and wing bending flexibility on gust alleviation. It has been shown that the above effects are significant for large flexible airplanes.[6,7] Some of the more recent techniques involve an improved mathematical description of an aircraft to 3-dimensional isotropic turbulence by establishing a unique correspondence between the spatial symmetry properties of the ambient field of turbulence and that of the aircraft thereby yielding a more elegant and compact formulation offering significant advantages over the direct multiple input approach. The new formulation utilizes four describing functions which depend upon the gust frequency and the interpanel separation, to determine the normalized gust velocity cross spectra for the turbulence field[8]. [9]Eigenspace techniques have been used for the design of an active flutter suppression / gust load alleviation system for a hypothetical research drone. Full state control laws were designed

for the above system by selecting feedback gains which place closed loop eigenvalues and shape the closed loop eigenvectors so as to stabilize wing flutter and reduce gust loads at the wing root while yielding acceptable robustness and satisfying the constraints on r m.s. control surface activity.

Interest in the gust response control has been resumed because airplanes have become more sophisticated and versatile in their component buildup that might be used to advantage for gust response control purposes without adding complexity or compromising the design. Further, the need is greater, more is known about the problem, and the subject now includes the idea of supressing the structural bending effects. Some of the more recent work also incude the concepts of controlling the elastic mode response of the airframe called mode stabilization or modal response control. It is neccessary for us to continually re-examine and improve our analytical and experimental modelling of the atmosphere for application to design and flight simulation. Moreover, the recent explosive growth in computing capability has provided unprecedented opportunities for sophisticated analysis, control and simulation.

1.3 Active Control Technology

Active control technology (ACT) involves continuous measurement of several flight variables via sensors installed at selected stations on the aircraft, complex processing of signals obtained from the sensors and feed them back to actuate various aerodynamic surfaces to achieve the required objective. ACT is applied in some very critical regimes of operation of aircraft involving their safety. Gust response and the subsequent alleviation of the loads and motions that result is one such area where ACT may be used very effectively because of problems of controllability and safety of the aircraft that arise upon gust encounter.

1.4 Preview of the work

Chapter 2 describes the characterisation of the gusts. It describes the various approaches that have been used so far to model the gusts. The use of the Von Karman gust model for the mathematical modelling of the gust for the present study has also been justified.

Chapter 3 discusses how the problem of gust alleviation has been formulated in terms of the equations of motion. It also shows the incorporation of a general control law – feedforward and feedback into the equations of motion. The signals from the feedforward loop are fed to the flap (direct lift controls) while the signals from the feedback loop are fed back to the elevator.

Chapter 4 elaborates the method of analysis used in computation and the use of power spectral density approach in the analysis.

Chapter 5 discusses the results obtained from computation.

Chapter 2. Characterisation Of Gusts

The influence of turbulence on aircraft flight has always been of concern, since the Wright Brothers because of the control upset problem, influence on the static design strength of the aircraft and problems in acheiving precise flight. The study of gust response of airplanes is a twofold problem requiring the adequate representation of the characteristics of atmospheric turbulence and the determination of the airplane response (loads or motions) in rough air. Trends in aeronautics towards higher speed and the development of missiles has served to increase the need for more generally applicable techniques, both for the measurements of the characteristics of atmospheric gusts and for the calculation of the gust response of new airplanes. Our concern is with the dynamic features of the problem - the response of the airplane associated with the gradients in atmospheric motion which are termed gusts.

2.1 Mathematical Modelling of Gusts

For many years, it was common practice to treat the gust response problem as response to discrete gusts considered to be isolated encounters with steep gradients (horizontal and/or vertical) in the horizontal or the vertical speed of air. The earlier gust studies were in terms of the concept of a single or discrete gust encounter where the gust shape was considered to be of the sharp edged or step function type. The airplane was supposed to be in steady flight in quiescent air and suddenly enter a region of updraft. The updraft may be sharp edged or graded. Later, this was modified to incorporate a specified profile of the gust. Figure 2.1shows the discrete gust approach to design which is still in partial use

today.

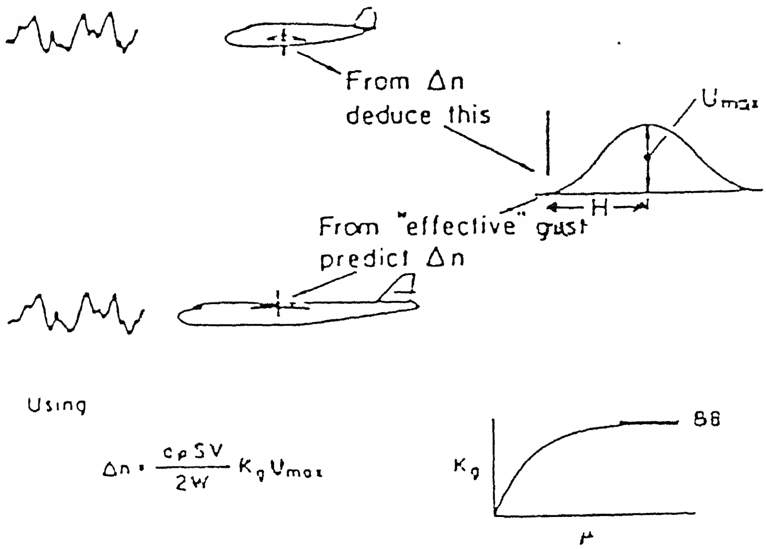


Fig 2.1. Discrete Gust Approach

The (1 - cosine) shape of the gust is one of the favoured forms in which W_g is the gust velocity and d is the the distance along the flight path.

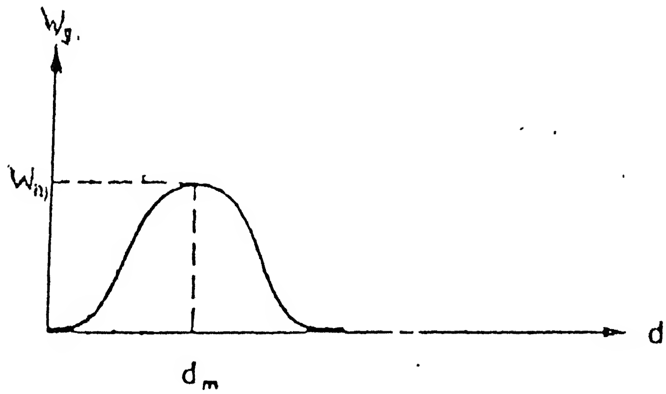


Fig 2.2. (1 — cos) gust shape

The severity of the gust is controlled by the magnitude of W_m and d_m.

In the light of random turbulence approach, discrete gusts are identified with the occasional extreme excursions of the upward velocity. In this sense, the spectral approach includes the discrete gusts, while at the same time giving a more

realistic and comprehensive picture. Nowadays, spectral techniques are being used for designing aircraft for gust encounter. The assumptions commonly made are :

(a) Taylor's hypothesis applies, namely that time histories of gust velocities obtained from aircraft may be converted to space fixed, temporarily 'frozen', space histories

(b) Gusts are considered to be random in the flight direction only but are considered uniform in the spanwise direction due to the notion that the scale of turbulence is large as compared to the airplane dimensions.

The basic spectral procedure involves choosing a spectrum for velocity input, a scale value L, an rms gust severity value σ_W and then through the frequency response to get the output response spectra as shown in figure 2.3.

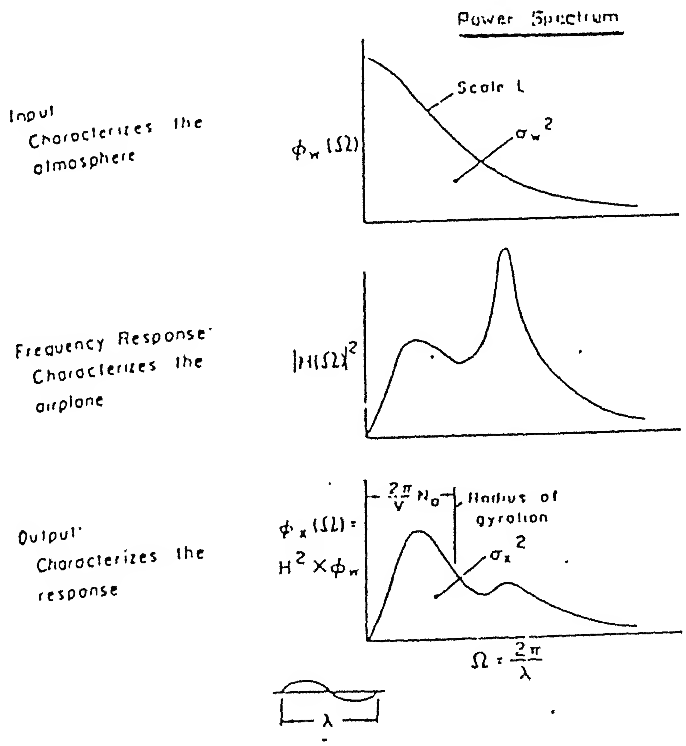


Fig 2.3. Input Output Relations for gust response

The frequency response function 'H' defined as the response of a variable ' x ' to

unit sinusoidal gust is fundamental to the spectral approach. Early statistical ·eatments of the atmospheric turbulence usually employed a spectral shape due to ·yden for mathematical modelling purposes. This spectrum, based on exponential ırve representations of the correlation functions for turbulence velocities, khibits an ω^{-2} fall off at high frequencies. More recent studies have popularly sed the spectrum equation due to Von Karman which exhibits an $\omega^{-\frac{5}{3}}$ fall off at ıgh frequencies. Figure 2.4 indicates that the Von Karman model of the atmosperic ırbulence is the preferred form for modelling purposes and will be used in the ·esent study.

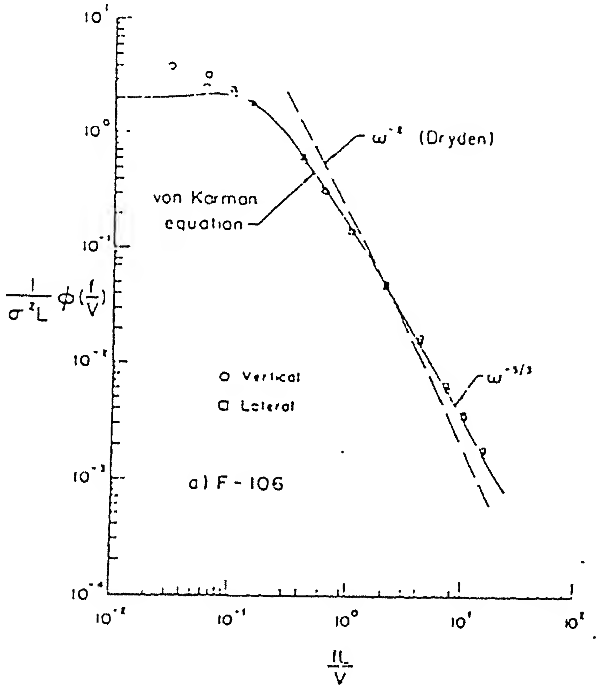


Fig 2.4. Spectral Shape of Atmospheric Turbulence

'he form adopted by Von Karman for $E(\Omega)$ is as below

$$E(\Omega) = C \frac{(\Omega/\Omega_0)^4}{[\ 1 + (\Omega/\Omega_0)^2]^{17/6}} \qquad 2.1$$

This exhibits a behaviour of Ω^{-4} and $\Omega^{\frac{-5}{3}}$ at low and high frequencies respectively. The constants C and Ω_0 and the one-dimensional spectra that are obtained from

this relation by using the relations in Appendix A are as below :

$$\phi_u = \sigma_u^2 \left(\frac{2L}{\pi}\right) \frac{1}{[1 + (1.339L\Omega)^2]^{5/6}} \qquad 2.2$$

$$\phi_w = \sigma_w^2 \left(\frac{L}{\pi}\right) \frac{[1+\frac{8}{3}(1.339L\Omega)^2]}{[1 + (1.339L\Omega)^2]^{11/6}} \qquad 2.3$$

Specific items that are usually considered in the description of turbulence are the following:

(a) Proportion P of the total time that the turbulence exists

(b) Spectral shape $\phi(\Omega)$

(c) RMS turbulence severity values σ_u, σ_v, σ_w

(d) Integral scale length L

(e) Peak or level crossing count of turbulence time histories

(f) Isotropy and homogeneity checks

(g) Effect of atmospheric variables

The influence of geographic location, season and altitude on the above is also of concern and needs to be considered.

Turbulence can be further classified into four categories on the basis of turbulence severity :

(a) Light

(b) Moderate

(c) Severe

(d) Extreme

A common means of representing severity of turbulence patches is in terms of the rms value (σ) of the velocities sensed. The rms value of turbulence, using the data on the proportion of time that the patches of various severity are found,

appears to be essentially invariant with the altitude and is in the range of 3–3.5 fps. Atmospheric turbulence is a random process in which the maximum values of the gust velocities are of the order of 20 times the overall rms value.

Turbulence severity tends to be linearly related to wind speed U in the form

$$\sigma_W = \sigma_{W0} + m\,U \qquad 2.4$$

where σ_{W0} is dependent on atmospheric stability (increasing turbulence with decreasing atmospheric stability) and the slope m appears to be larger for rough terrains than for the plains.

Also intimately associated with the severity and the spectral shape of turbulence data is the integral scale of turbulence L, whose mean value lies in the range of 500–700 ft. Figure 2.5 shows that though the scatter is great, but a rough trend indicates that the scale length increases as the severity increases. Press and Meadows[19] have suggested a value of 1000 ft for L as representative of atmospheric turbulence. The worst-case philosophy as specified by the airworthiness requirements suggests a value of σ = 10 ft/s.

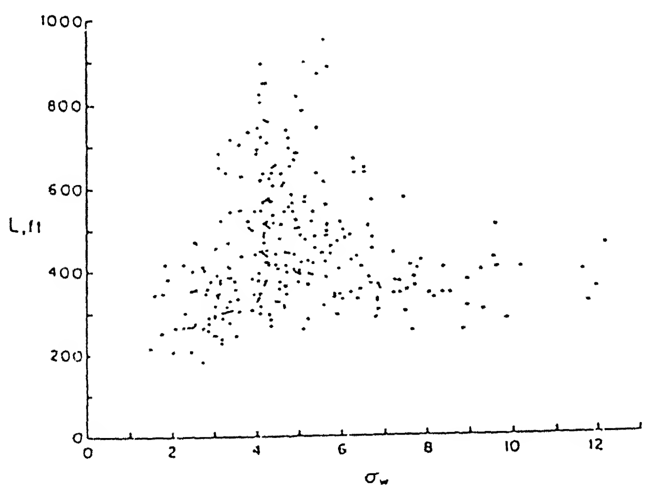


Fig 2.5. Correlation of Integral Scale L with rms gust severity σ

2.2 Simplifications

An assumption commonly made is that atmospheric turbulence is isotropic meaning that there is no mean rate of transfer of momentum across shearing surfaces or Reynolds or eddy shearing stresses, $-\rho\overline{uv}$, $-\rho\overline{uw}$, $-\rho\overline{vw}$ vanish and the cross spectra ϕ_{uv}, ϕ_{uw}, ϕ_{vw} also vanish where u, v, w represent longitudinal, lateral and vertical turbulent velocities. Taylor's hypothesis applies, namely that time histories of gust velocities obtained from aircraft may be converted to space fixed, temporarily 'frozen', space histories. Also, atmospheric turbulence may be regarded as a momentarily time frozen random surface in space relative to an airplane ie. the properties of stationarity and homogeneity of atmospheric turbulence are also assumed. Statistical functions such as the auto correlation function in terms of either time or distance is thus made possible through the transformation $x = Vt$ due to Taylor's hypothesis. Some key relations applying to isotropic turbulence are listed in Appendix A.

The overall response for a completely random gust environment is smaller than the case when only the random nature of the gust in the flight direction is considered. Essentially, the above problem may be treated as that of a system being excited by multiple random inputs rather than by a single one. Cross spectra between the individual inputs must now be considered along with the spectral value for each input. The random velocity of the atmosphere is used as input to the airplane. Because of the linearity of the system (the airplane), the three component velocities of the input can be treated separately. The general gust response problem is sub-divided into three subcases : the response of the airplane to (a) fore-and-aft gusts u_g, (b) side gusts v_g and (c) up-and-down or vertical gusts w_g. Generally, longitudinal gusts may be ignored. Wing loads are primarily governed by vertical gusts. Fuselage and tail loads can be obtained by superimposing results that are obtained in separate treatments for lateral and vertical gust encounter.

Each of these may induce technically important airplane response, although it is the vertical component w_g that is mainly responsible for normal accelerations which is the main interest of the present study. The analysis of longitudinal response to vertical gust requires only a one-dimensional gust spectrum because the linear component of the spanwise variation of the vertical gust w_g has no effect on the longitudinal motion. Unsymmetrical vertical and side gusts generate rolling & yawing moments. They are also generally specified by the airworthiness requirements.

Chapter 3. Formulation Of the Problem

The treatment of the gust response problem of the aircraft needs a precise formulation of the equations of motion. The equations of motion refer to a space fixed coordinate system because the gust vel is only defined relative to the ground. The longitudinal perturbation equations of motion in steady rectilinear flight of a conventional aircraft under the usual assumption of small perturbations and linearity (of perturbed force and moment coefficents as a function of perturbed variables of motion) are well known and may be found in standard textbooks on Dynamics of Flight[14-15] .

3.1 Equations of motion

The equations of motion with reference to stability axes are given below in dimensional form :

$$\dot{u} = X_u\, u + X_{T_u}\, u + X_\alpha\, \alpha + g \cos\theta_e\, \theta + X_\eta\, \eta \tag{3.1}$$

$$U_e\, (\dot{\alpha} - q) = Z_u\, u + Z_\alpha\, \alpha + Z_{\dot{\alpha}}\, \dot{\alpha} + Z_q\, q - g \sin\theta_e\, \theta + Z_\eta\, \eta \tag{3.2}$$

$$\dot{q} = M_u\, u + M_{T_u}\, u + M_\alpha\, \alpha + M_{T_\alpha}\, \alpha + M_{\dot{\alpha}}\, \dot{\alpha} + M_q\, q + M_\eta\, \eta \tag{3.3}$$

$$\dot{\theta} = q \tag{3.4}$$

where U_e and θ_e are the airspeed and the pitch angle in the reference flight condition respectively; u, α, q, θ and η are the perturbation velocity, angle of attack, pitch rate, pitch attitude and the elevator angle respectively. The

dimensional derivatives are defined in Appendix B.

The following simplifications in the equations of motion may be introduced. The reference flight condition is a stationary horizontal flight path ($\theta_e = 0$). Also the thrust T is assumed to be constant so that the derivative M_{T_u} and M_{T_α} may be assumed to be of the order zero. The above equations of motion are non dimensionalised and take the form as below.

$$(2\mu D - 2C_{L_e}\tan\theta_e - C_{x_u})\hat{u} - C_{x_\alpha}\alpha + C_{L_e}\theta - C_{x_\eta}\eta = 0 \qquad 3.5$$

$$(2C_{L_e} - C_{z_u})\hat{u} + (2\mu D - C_{z_{\dot\alpha}}D - C_{z_\alpha})\alpha - [(2\mu + C_{z_q})D - C_{L_e}\tan\theta_e] \theta - C_{z_\eta}\eta = 0 \qquad 3.6$$

$$-C_{m_u}\hat{u} - (C_{m_{\dot\alpha}}s + C_{m_\alpha})\alpha + (i_B s^2 - C_{m_q}s)\theta - C_{m_\eta}\eta = 0 \qquad 3.7$$

$$\hat{q} = D\theta \qquad 3.8$$

where $\mu = \frac{m}{\rho S l}$ $\qquad t^* = \frac{l}{u_0}$ $\qquad l = \frac{\bar{c}}{2}$ $\qquad i_B = \frac{I_{yy}}{\rho S l^3}$

The non dimensionalized quantities are defined in Appendix B. The relations between the non dimensionalized quantities and the longitudinal stability derivatives are also given in Appendix B.

However, for an aircraft provided with pitch attitude feedback to the elevator, the equations of motion have to be suitably modified. It may be noted that the feedback to the elevator is not the absolute value of the pitch attitude but the perturbation value. The above accounts to the addition of effective axial and normal force derivatives C_{x_θ} and C_{z_θ} and an effective moment derivative C_{m_θ} to the equations of motion. But, since the effect of the elevator on the pitching of the aircraft is significantly more predominant in comparison to the axial and the

normal force derivatives, C_{x_θ} and C_{z_θ} may be ignored. The term $+C_{m_\theta}\,\theta$ is added to the right hand side of the pitching moment equation.

3.2 Idealization of the airplane

The airplane may be idealized with varying degrees of approximation corresponding to the various assumptions about the way in which it feels the gust structure. The simplest approximation is to treat the airplane as a point which flies along a straight line. This, evidently, is not a very good approximation. The next step of elaboration is to represent the airplane as a segment of x-axis. In this case, the variation of the vertical gust in the longitudinal direction is considered but the spanwise and the vertical variation is still neglected. This is quite a good approximation of the airplane for the present study because normal accelerations are mainly due to the vertical component w_g. Reduction of normal accelerations is the main criterion to improve passenger comfort. Besides, the analysis of the longitudinal response requires only a one-dimensional gust spectrum because the linear component of the spanwise variation of w_g has no effect on the longitudinal motion. In the case of rigid body motion, no fast motions occur. Also, in a natural (random) gust field, the low frequency gusts are dominant. Gusts of higher frequency appear with small amplitudes as can be seen by the well known Dryden or Von Karman gust spectra. In addition, gusts having a smalll wavelength as compared with the length of the airplane cannot excite the motion of the airplane because they cancel each other. Thus, spectral components of wavelengths larger as compared to the length of the airplane l (ie. $\lambda \geq 8\ l$) are considered where a straight line is a fair approximation to most segments of a sine wave. For relatively long wavelengths, the variation in w_g along the length of the airplane is approximately linear so that the aerodynamic effect of the gust is to modify the effective angle of attack and pitch rate. The aerodynamic effect of the gradient

in u_g is equivalent to a certain rate of pitch. By incorporating aerodynamic force terms corresponding to α_g and $\hat{q}_g$ into the equations of motion, the effects of long wavelength gust components on the airplane response is adequately accounted for

3.3 Modified Equations of Motion

The equations of motion for controls fixed and a horizontal reference flight path after taking the Laplace transform are as follows :

$$(2\mu s - C_{x_u})\bar{u} - C_{x_\alpha}\alpha + C_{L_e}\bar{\theta} = -C_{x_\alpha}\frac{\bar{w}_g}{u_o} \qquad 3.9$$

$$(2C_{L_e} - C_{z_u})\bar{u} + (2\mu s - C_{z_{\dot\alpha}}s - C_{z_\alpha})\alpha - (2\mu + C_{z_q})\bar{\theta} = (-C_{z_{\dot\alpha}}s - C_{z_\alpha} + C_{z_q}s)\frac{\bar{w}_g}{u_o} \qquad 3.10$$

$$-C_{m_u}\bar{u} - (C_{m_{\dot\alpha}}s + C_{m_\alpha})\alpha + (i_B s^2 - C_{m_q}s)\bar{\theta} = (-C_{m_{\dot\alpha}}s - C_{m_\alpha} + C_{m_q}s)\frac{\bar{w}_g}{u_o} \qquad 3.11$$

where the bar indicates the Laplace transform of the quantities.

Since direct lift control (flap) and elevator control is being used for gust alleviation, the control derivatives appear on the right hand side of the above equations as :

$$\ldots\ldots\ldots \qquad = C_{x_\eta}\bar{\eta} + C_{x_\delta}\delta_{flap} \qquad 3.12$$

$$\ldots\ldots\ldots \qquad = C_{z_\eta}\bar{\eta} + C_{z_\delta}\delta_{flap} \qquad 3.13$$

$$\ldots\ldots\ldots \qquad = C_{m_\eta}\bar{\eta} + C_{m_\delta}\delta_{flap} \qquad 3.14$$

As pointed out earlier in Chapter 1, a mixed open loop and closed loop system

might be promising for small transport aircraft. The open loop portion performs a gross alleviation while the closed loop portion is responsible for the suppression of residual motions. A general control law was incorporated having an open loop control law which presupposes an exact measurement of the gust angle of attack signal and a closed loop feedback law which incorporates all the motion variables through suitable gains.

$$\left[\delta_{flap}\right]_{ol} = K_{\alpha_g}\,(1 + \tau s)\,\bar{\alpha}_g \qquad 3.15$$

$$\bar{\eta}_{cl} = K_u\,\bar{u} + K_\alpha\,\alpha + K_q\,\bar{q} + K_\theta\,\theta + K_n\,\bar{n} \qquad 3.16$$

where $\bar{n} = \dfrac{2\,u_0^2}{g\,c}\,(\bar{q} - s\,\bar{\alpha})$

Thus, $\bar{\eta} = \bar{\eta}_{cl} + \bar{\eta}_{pilot}$ 3.17

The above is incorporated into the equations of motion as below :

$$\begin{bmatrix} As+B & Cs+D & Es+F \\ G & Hs+I & Js+K \\ L & Ms+O & Ps^2+Qs+R \end{bmatrix} \begin{bmatrix} \bar{u} \\ \alpha \\ \bar{\theta} \end{bmatrix} = \begin{bmatrix} Ts+U \\ Vs+W \\ Xs+Y \end{bmatrix} \left[\frac{\bar{w}_g}{u_0} \right] + \begin{bmatrix} E1 \\ E2 \\ E3 \end{bmatrix} \left[\bar{\eta}_{pilot} \right]$$

where A, B, - - - - - - - - - are defined in Appendix B.

The above formulation of the equations of motion for the gust response problem is used for computation. The ' s ' operator of Laplace transform is written as ' ik ' where k (= Ωl) is the non dimensional frequency.

Chapter 4. Method Of Analysis

The study of gust loads on airplanes is a twofold problem requiring the adequate representation of the characteristics of atmospheric turbulence and the determination of the airplane response (loads or motions) in rough air.Trends in aeronautics towards higher speed and the development of missiles has served to increase the need for more generally applicable techniques, both for the measurements of the characteristics of atmospheric gusts and for the calculation of the gust loads on new airplanes. Turbulence can be defined as the irregular random motion of small fluid masses which is so complex that there can be no hope of a theory which will describe in detail the velocity and the pressure fields at every instant. Existing theories for modelling atmospheric turbulence may be classified as either empirical or statistical.

4.1 Theory of Generalized Harmonic Analysis

The theory of generalized harmonic analysis[6] appear adaptable for extending the analysis of gust loads beyond the discrete gust case to the case of continuous turbulence. Techniques from generalized harmonic analysis involving the power spectral density have been used in diverse fields for many years, such as in the study of random noise problems in communications and in the study of small scale turbulence in wind tunnels. Attractive features of spectral analysis for the study of gust loads are the possibiities that :

(a) Continuous turbulence can be described in analytic form by a power spectrum rather than by discrete gusts.

(b) The load response of the airplanes to continuous rough air can be evaluated.

(c) The desirable response characteristics of an airplane for minimizing gust effects in continuous rough air will become amenable to analysis.

The theory of harmonic analysis is described in Appendix C.

The response of the airplane to rough air is essentially a statistical one ie. the power spectrum of the input (the air turbulence) is known and the power spectrum of the output (the airplane response) is required to be found out. From the theory of spectral analysis defined above, the link which connects the input and the output is the square of the modulus of the transfer functions of the motion variables and the gust velocity. The transfer functions obtained have been validated by using the numerical example of Sec 10.7 of [15]Etkin. The input output relation in terms of the space frequency Ω ($= \frac{\omega}{V}$) is :

$$\phi_o(\Omega) = \phi_i(\Omega) \quad | T(i\Omega) |^2 \qquad 4.1$$

Only the square of the modulus is required, the phase angle of the response is of no interest in this connection. The Von Karman model of the gust is being used to give us the power spectral density of the input gust spectrum. The longitudinal stability derivatives of a Roshkam's business jet transport aircraft[20] given in Appendix D are being used for studying the aircraft response to gusts. The data for the flap used is also given in Appendix D.

4.2 Passenger Comfort

The reduction of accelerations caused by rough air would be of obvious value for improving passenger comfort in commercial airline operation. An airplane capable of smooth flight through rough air would also be a valuable tool for studying the gust structure of the atmosphere. The design of a device for providing comfortable flight to the passengers of an airplane in rough air requires a

knowledge of the factors which contribute to passenger comfort. [17]McFarland indicates that slow oscillations of large amplitude are more likely to cause sickness than faster oscillations of small amplitude. Some of the results obtained in these tests are shown in figure on the next page.

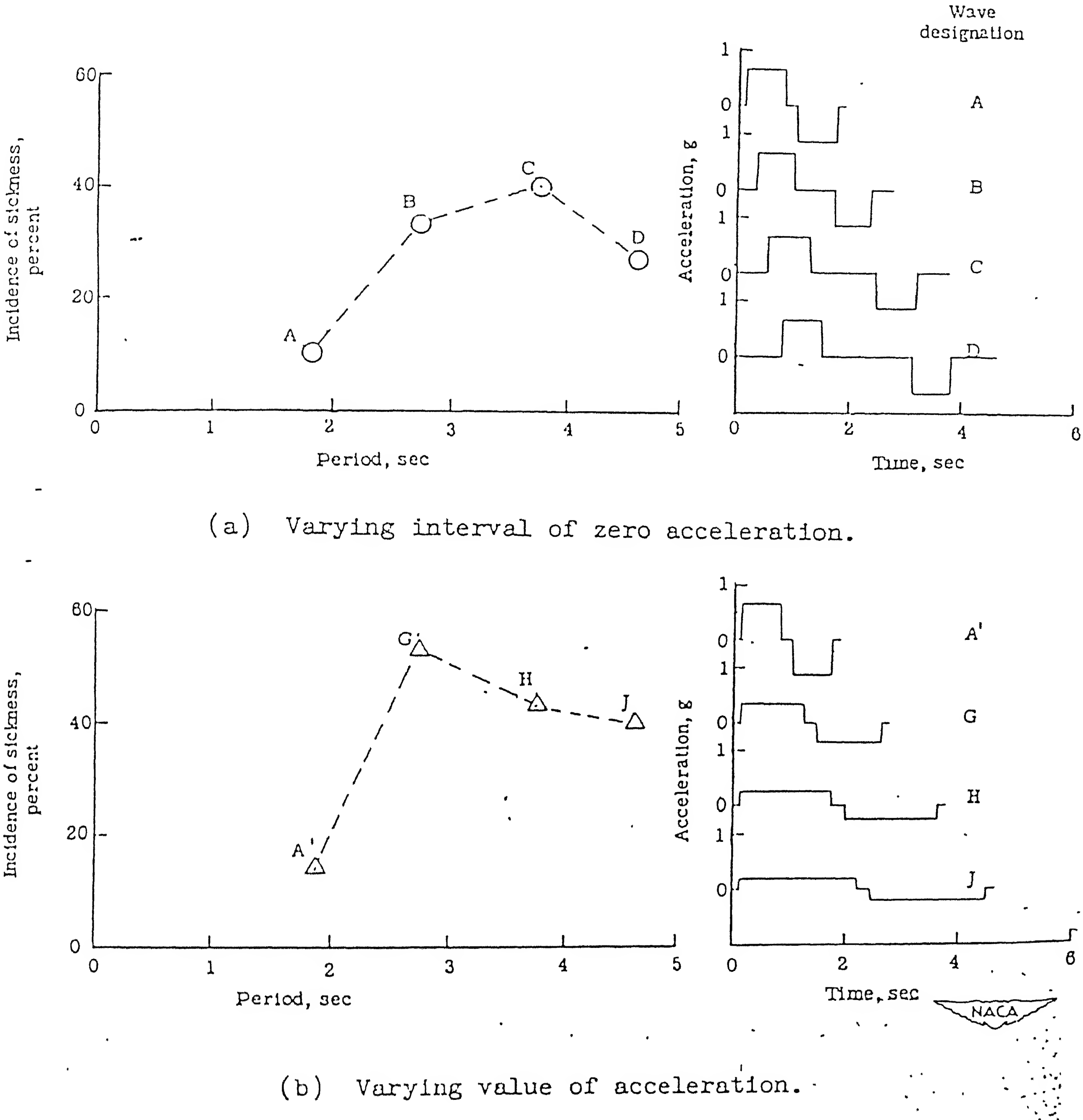


Figure 4.1 Percentage of men who became sick within a period of 20 minutes when subjected to vertical oscillations of various periods and wave forms. Various symbols indicate different series of tests. Primes designate repeat tests. Results obtained from reference 2.

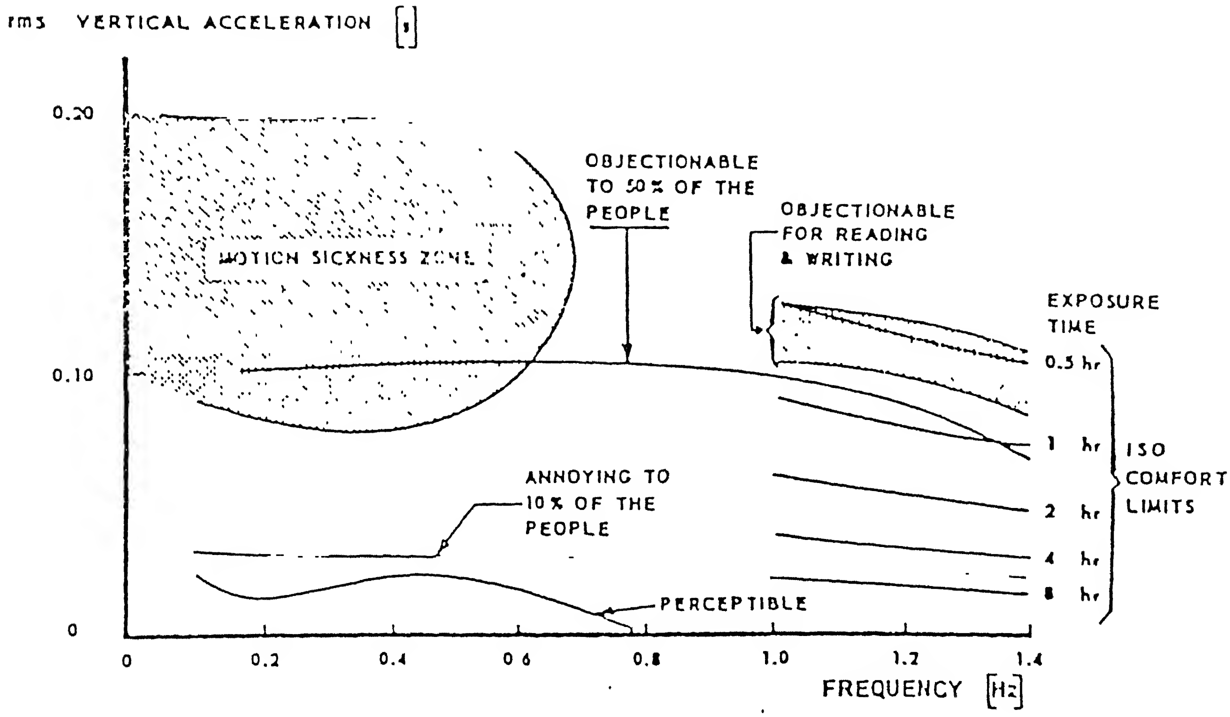


Fig. 4·2 Tentative passenger comfort boundaries in the low frequency range

The results of these tests showed that very little sickness was produced by the shortest period oscillation tested and this appears to be in accord with the common experience eg. a periodic motion of a small boat in a rough sea is known to produce sickness in a large number of passengers, whereas the motion of a trolley car on a rough track, which involves abrupt jolts and jerks containig components of oscillation of high frequency and fairly large amplitude, causes sickness in very few passengers. Reduction of the more prolonged changes in vertical acceleration, thus appears to be beneficial for minimizing sickness.

Other stimuli that may be important in producing motion sickness include lateral or rotational accelerations, motion of objects in the field of vision, and many psychological factors such as noise, vibration, temperature, ventilation and so forth. Thus, even complete elimination of normal accelerations would not neccesarily eliminate airsickness. In flight through rough air, the changes in normal acceleration are relatively large, whereas lateral accelerations, rotational accelerations, and changes in orientation are relatively small. Peroidic changes in normal acceleration are a major cause of motion sickness and its reduction thus appears to be the most promising method of improving passenger comfort. Motion sickness occurs only at low frequencies (0.1 – 0.4 Hz) in connection with large acceleration amplitudes. For the present study, the improvement of passenger ride comfort is the main aim of gust alleviation. In order to measure the efficiency of the gust alleviation system, the ride comfort criterion is expressed in terms of the motion parameters. This can be accomplished by definining a comfort criterion. The main parameter influencing passenger comfort is the vertical acceleration. The root mean square (r.m.s.) value of the vertical acceleration is given by

$$C = \ddot{z} = \sqrt{\sigma^2_{\ddot{z}}} \qquad 4.2$$

In the time domain,

$$\sigma^2_{\ddot{z}} = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} \ddot{z}^2 (t) \, dt \qquad 4.3$$

where T is the time of observation.

In the frequency domain,

$$\sigma^2_{\ddot{z}} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi_{\ddot{z}} (\omega) \, dw \qquad 4.4$$

where $\phi_{\ddot{z}} (\omega)$ is the power spectral density of the vertical acceleration.

The above equations represents the simplest of the comfort criteria. In terms of the above index of passenger comfort, reduction of the root mean square (r.m.s.) values of the vertical acceleration at various locations of the aircraft cabin would improve passenger comfort.

4.3 Computational Procedure

The study of gust response of aircraft in longitudinal motion has been done computationally as follows :

(i) Computation of the roots of the characteristic equation.

(ii) Computation of the time responses of the motion variables like u, α, q, θ and load factor n.

(iii) Computation of the frequency response (power spectra) of the motion variables through the power spectral density approach.

(iv) Reduction of the r.m.s. value of normal accelerations through the proposed control law by optimising the control law variables (DownHill Simplex Method[20]) to improve passenger ride comfort.

(v) Includes the idea of a random gust by simulation of a random gust through a

control law by optimising the control law variables (DownHill Simplex Method[20]) to improve passenger ride comfort.

(v) Study the effect of relaxation of static stability on the gust alleviation of the aircraft[13].

Chapter 5. Discussion Of Results

This chapter discusses the results of the computations presented in the tables and figures. The effect of various feedforward and feedback laws is discussed. An exploratory study into the effect of relaxation of static stability and its artificial compensation or superaug_mentation through pitch attitude feedback to the elevator is discussed. A comparative study of the various control laws obtained by the minimization of the objective function (passenger comfort index) w.r.t. the various gains for the controls to be applied, is made through Table 5.1.

Table 5.1. Comparative Study Of The Various Control Laws

	Control Law Gains	Comfort index	Short Period Frequency (Hz)	Short Period t_{half} (sec)
1	Gust alleviation system disengaged	2.17146	2.65	0.685
2	K_{α_g} = 5.250667 τ_{filter} = 358.006165	2.03345	2.65	0.685
3	K_θ = 0.721041	1.77057	4.3573	0.842
4	K_u = 4.503037 K_α = 0.697434	1.65529	4.4185	0.633
5	K_{α_g} = 17.842354 τ_{filter} = 83.386169 K_α = 0.134872	1.63278	3.065	0.682
6	K_α = 0.057423	2.05263	2.8341	0.684

Fig 5.1 shows the power spectrum of the atmospheric turbulence as obtained from the Von Karman model of the gust as a function of the space frequency $\Omega = \frac{\omega}{V_e}$.

Fig 5.2 compares the various gust alleviation systems in terms of the reduction of load factor. Since the gust alleviation system affects only the short period motion of the airplane , we shall consider reduction of the load factor near the short period frequency. Refer Table 5.1 for the explanation below.

CASE 1 : When the gust alleviation system is disengaged, the r.m.s. value of the load factor (a measure of passenger comfort) is 2.17146. The conventional airplane has been assumed to have good handling qualities.

CASE 2 : When the gust alleviation system 2 (a feedforward loop which feeds the gust angle of attack α_g to the flap after passing it through a proportional plus derivative filter) is engaged, the r.m.s. value of the load factor reduces to 2.03345 (ie. 6.5%) which is not very significant. The frequency and the time to halve the oscillations of the short period remain the same. This means that there is no change in the handling qualities because damping of the short period mode is a crude estimate of the handling qualities.

CASE 3 : When the gust alleviation system 3 (pitch attitude θ feedback to the elevator) is engaged, there is a greater reduction in the r.m.s. value of the load factor than Case 2, a reduction of 18.6% which is quite significant with reference to our main aim of improving passenger comfort or alternatively, decreasing the r.m.s. value of the load factor. The frequency and t_{half} of the short period mode increases to 4.3573 Hz and 0.842 sec respectively. The increase in the short period frequency is helpful for improving passenger comfort because low

frequency oscillations are mostly responsible for causing motion sickness in most passengers. The decrease in short period damping indicates a deterioration in handling qualities.

CASE 4 : When the gust alleviation system 4 (forward velocity u and angle of attack α feedback to the elevator) is engaged, a reduction of 23.88% in the r.m.s. value of the load factor is achieved. The frequency of the short period mode increases to 4.4185 Hz while t_{half} decreases to 0.633 sec. The increase of the short period frequency is helpful for improving passenger comfort. The increase in damping of the short period mode is not very much and therefore not much deterioration in the handling qualities. Thus, this gust alleviation system appears to be promising towards improvement of passenger comfort.

CASE 5 : When the gust alleviation system 5 (a combination of the feedforward and feedback loop – the gust angle of attack α_g is directly fed to the flap after passing the signal through a proportional plus derivative filter and feedback of α to the elevator) is engaged, a reduction of 24.91% in the r.m.s. value of the load factor is achieved. The frequency of the short period mode increases to 3.065 Hz while t_{half} decreases to 0.682 sec. The shift of the short period towards the high frequency regime helps in improving passenger comfort. The increase in short period damping is very negligible and therefore no deterioration in the handling qualities. From the above, this gust alleviation system is more promising towards improvement of passenger comfort without the deterioration of handling qualities.

CASE 6 : When the gust alleviation system 6 (angle of attack α feedback to the elevator) is engaged, the r.m.s. value of the load factor reduces to 2.05263, a reduction of 5.6% which is an insignificant improvement in passenger comfort as

compared to Case 5. The frequency and t_{half} of the short period mode increases to 2.8341 Hz and 0.684 sec respectively.

Fig 5.3 shows that significant changes in forward speed are all associated with the phugoid mode as expected since speed changes are very small in the short period mode.

Fig 5.4 shows that the α response to gusts is flat except for a rise at the short period after which it falls of rapidly. The α response of the gust alleviation system 5 shows that the response is lesser than when the gust alleviation system is disengaged.

Fig 5.5 shows the pitch rate response. The response of the pitch rate becomes lesser when the gust alleviation system 5 is engaged than the case when gust alleviation system 4 is engaged or the system disengaged.

The load factor response is plotted in Fig 5.6 The load factor reduces when the gust alleviation system 5 is engaged and the r.m.s. value of the load factor reduces by almost 25% which significantly inproves the comfort of the passengers.

Fig 5.7 to Fig 5.10 show the output spectra obtained by multiplying the corresponding plots by the Von Karman gust input of Fig 5.1. Their shapes are similar to those of the reponse functions except that they fall off more rapidly at the high frequency end.

Fig 5.11 shows the time history of a random vertical gust simulated through random number generation[20-21] to have the power spectrum similar to the Von Karman gust.

Fig 5.12 shows the power spectrum of the random gust and compares it with the Von Karman gust model.

Fig 5.13 to 5.16 show the output time history of the motion variables for the random gust for the cases when the gust alleviation sytem is engaged and disengaged.

Fig 5.17 shows the gust alleviation system.

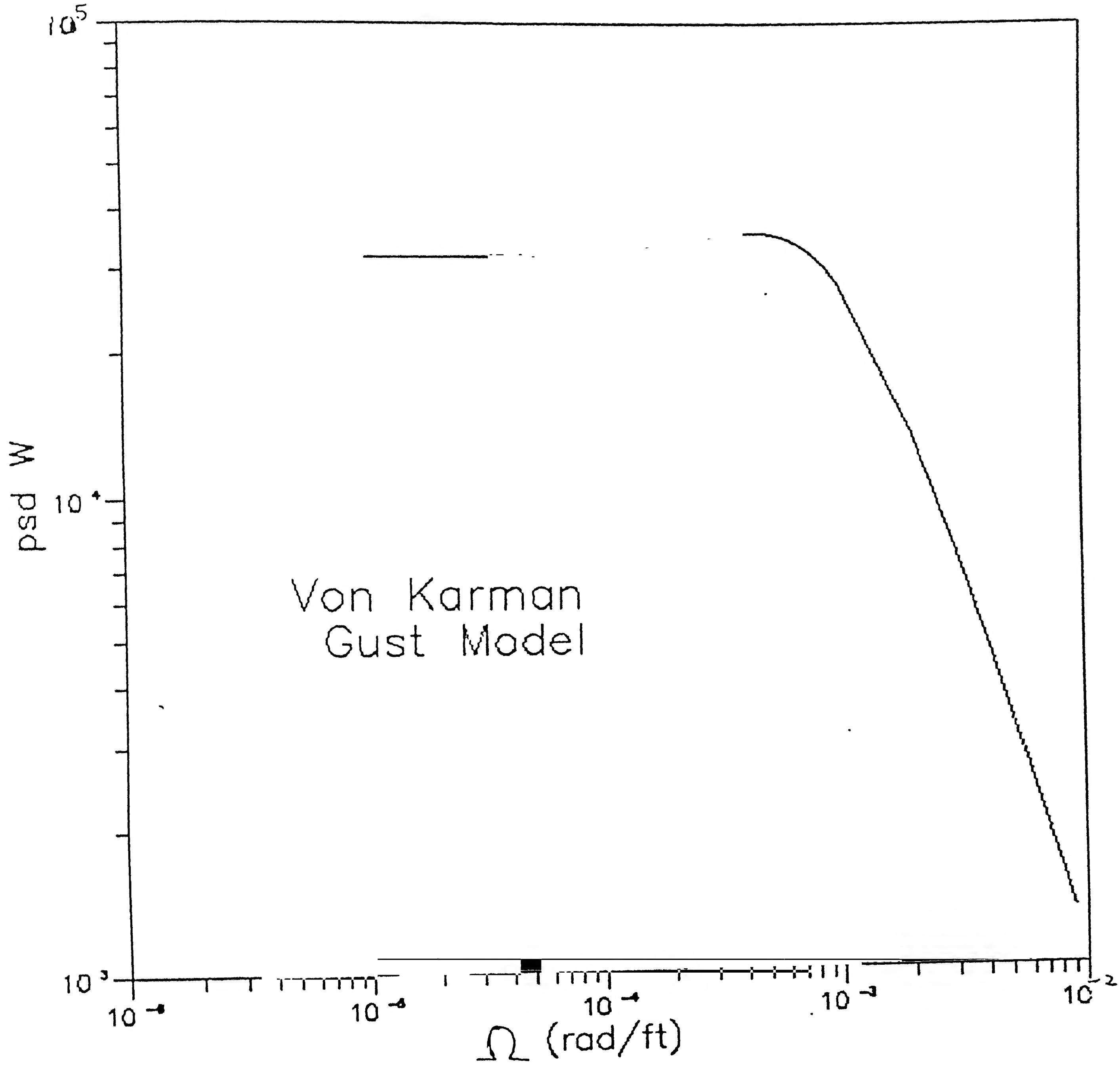
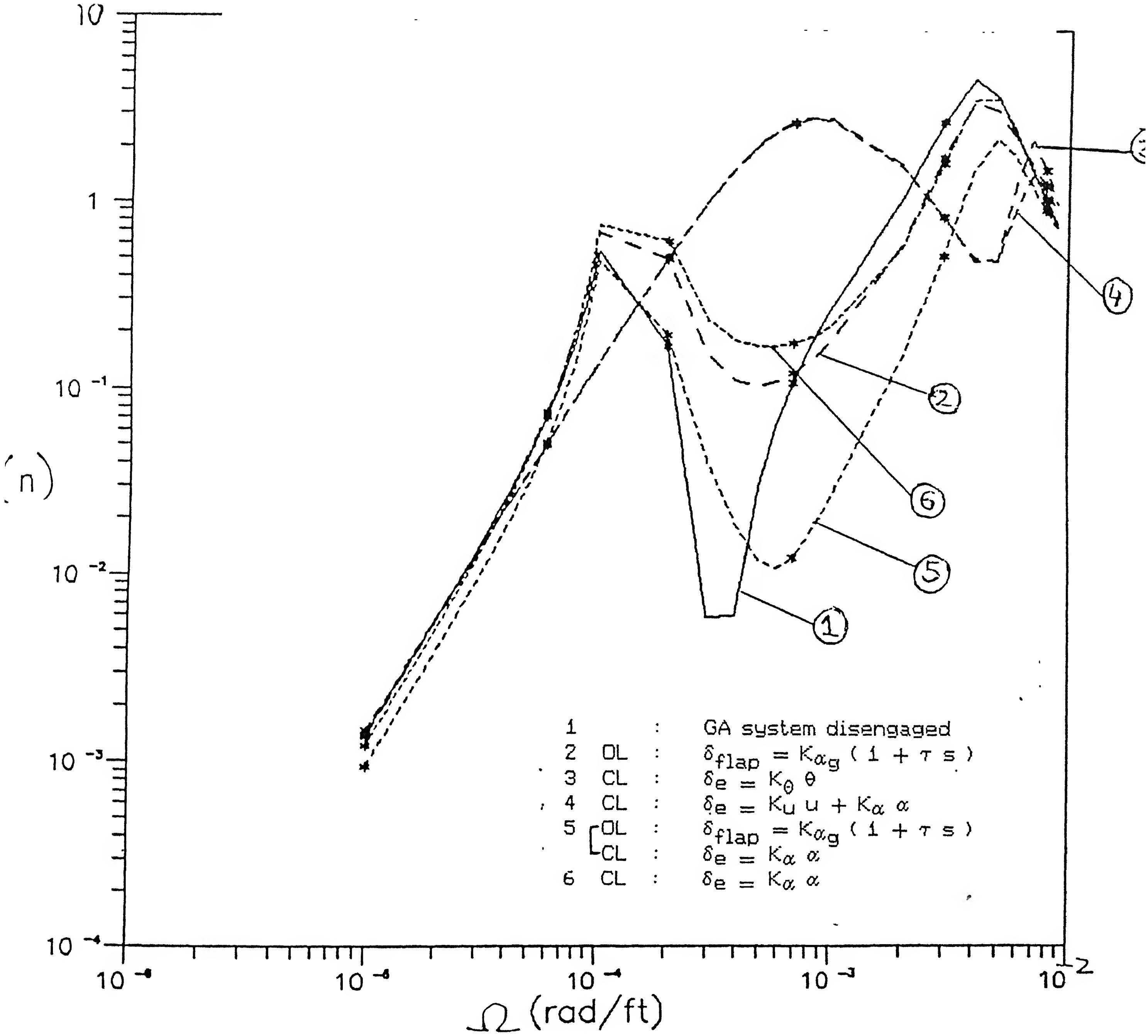
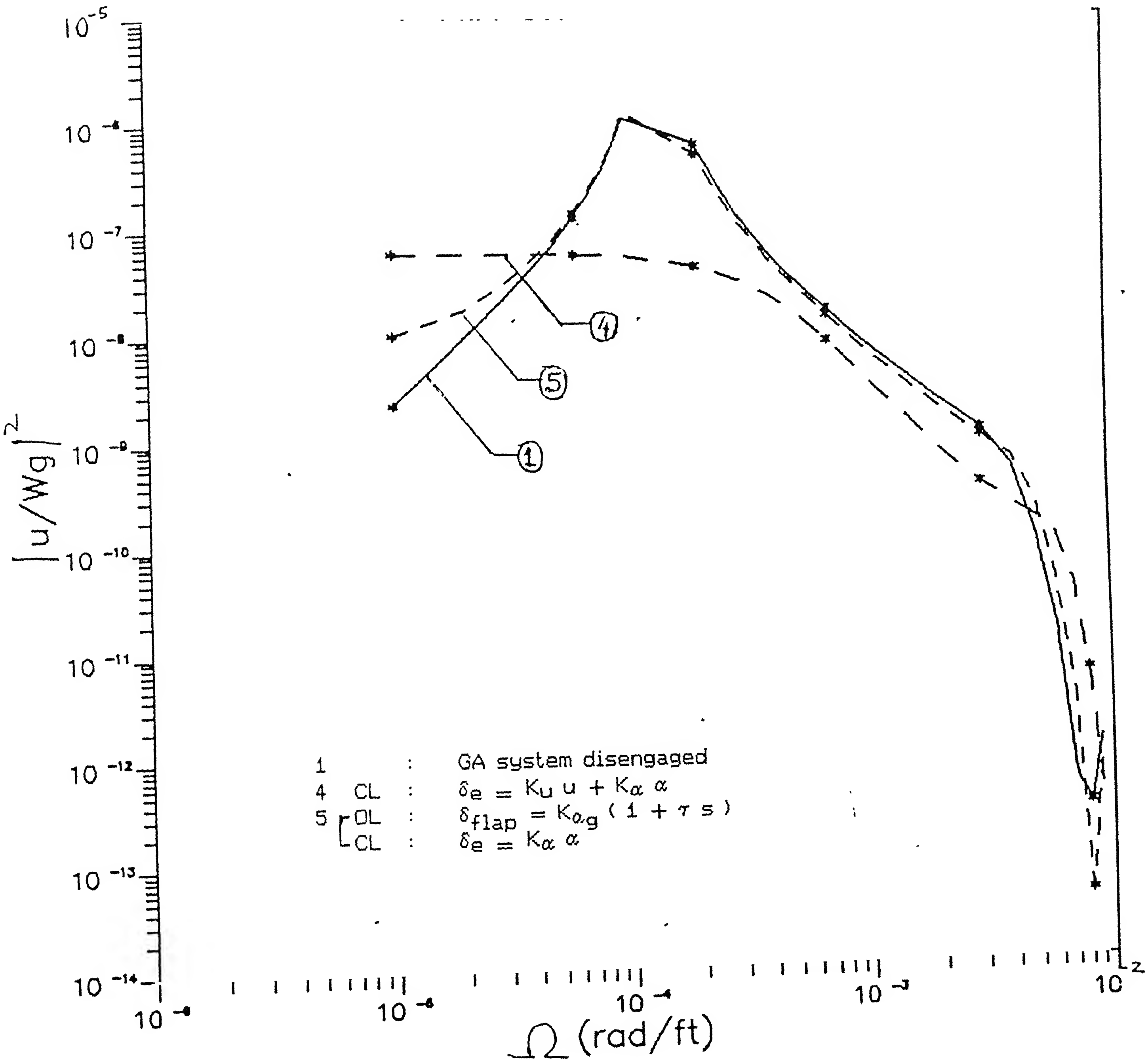


Fig 5.1 Power Spectrum of the Gust (Von Karman Mode



g 5.2 Power Spectrum of n



ig 5.3 Variation of the square of the modulus of TF (of u w.r.t. Wg) for the two promising control laws

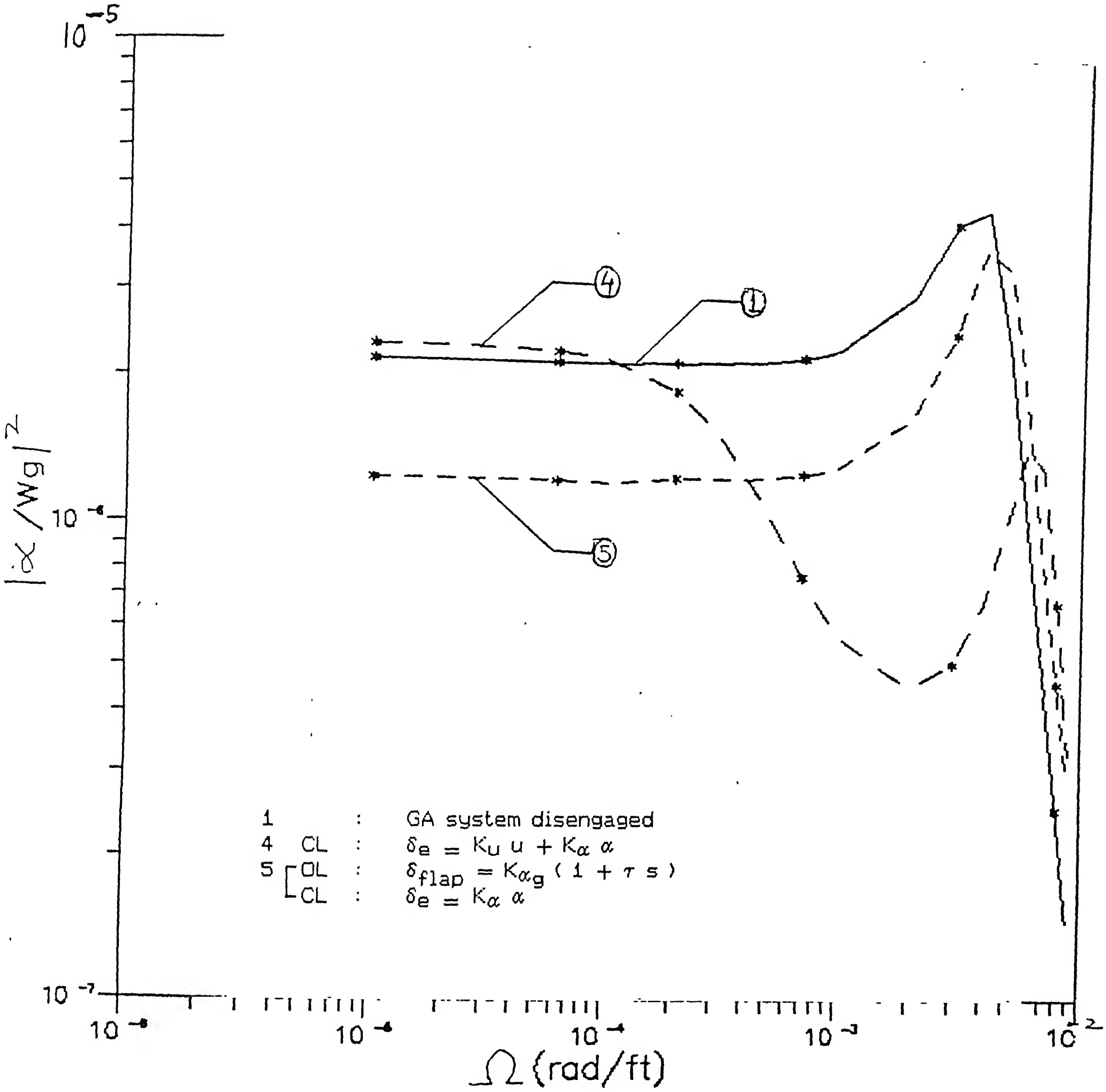


Fig 5.4 Variation of the square of the modulus of TF (of α w.r.t. Wg) for the two promising control laws

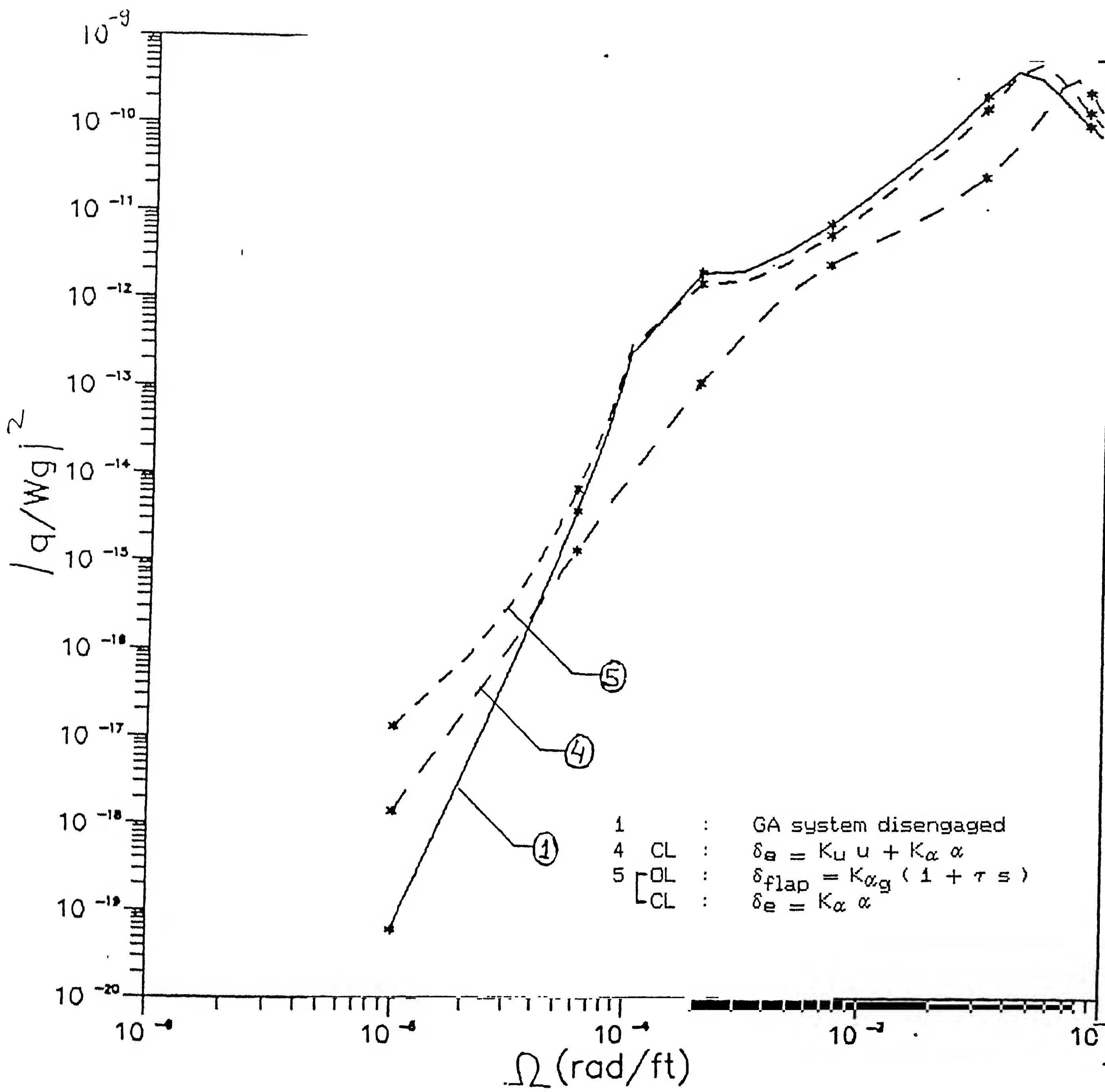


Fig 5.5 Variation of the square of the modulus of TF (of q, w.r.t. Wg) for the two promising control laws

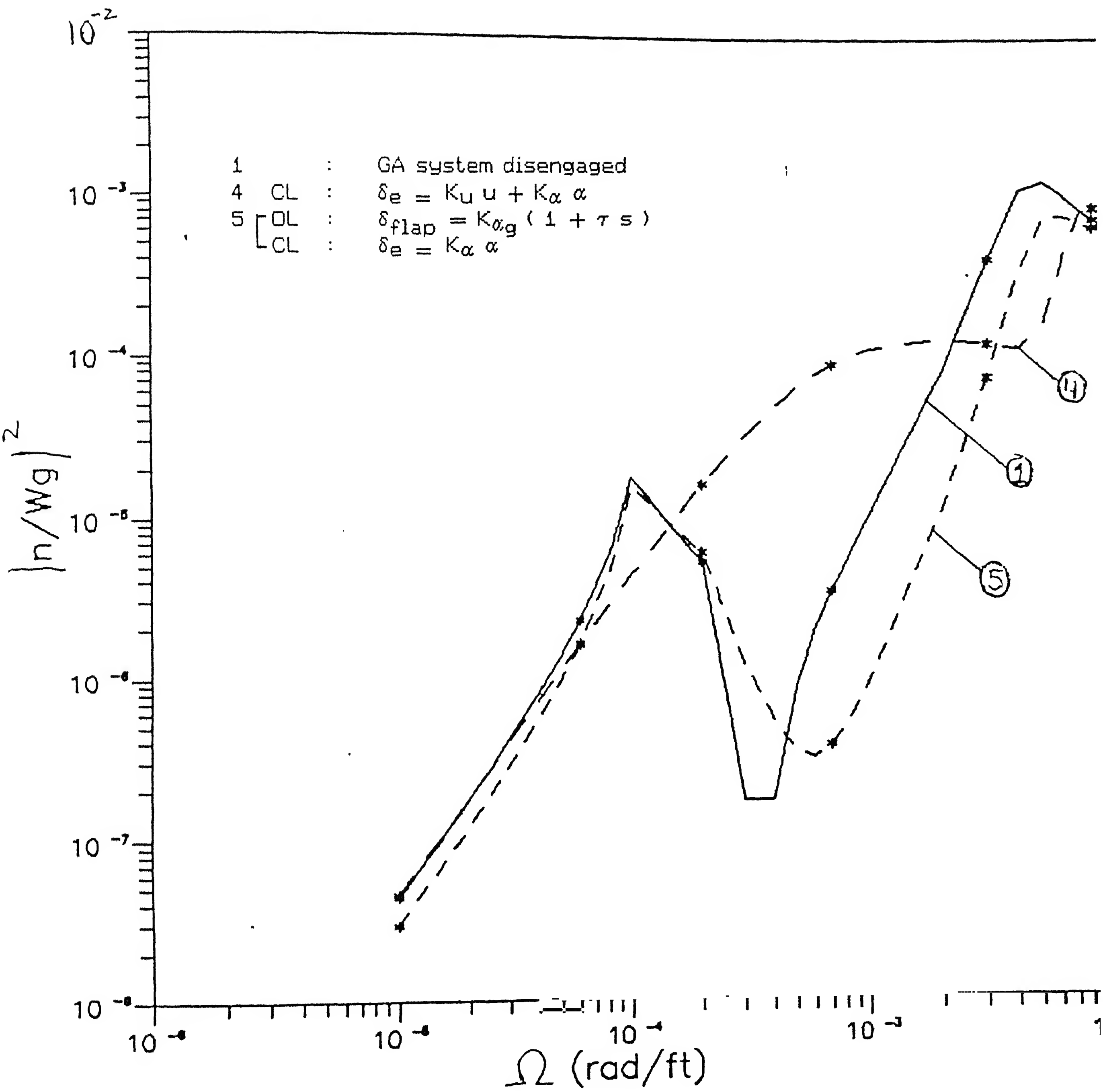


Fig 5.6 Variation of the square of the modulus of TF (of η w.r.t. Wg) for the two promising control laws

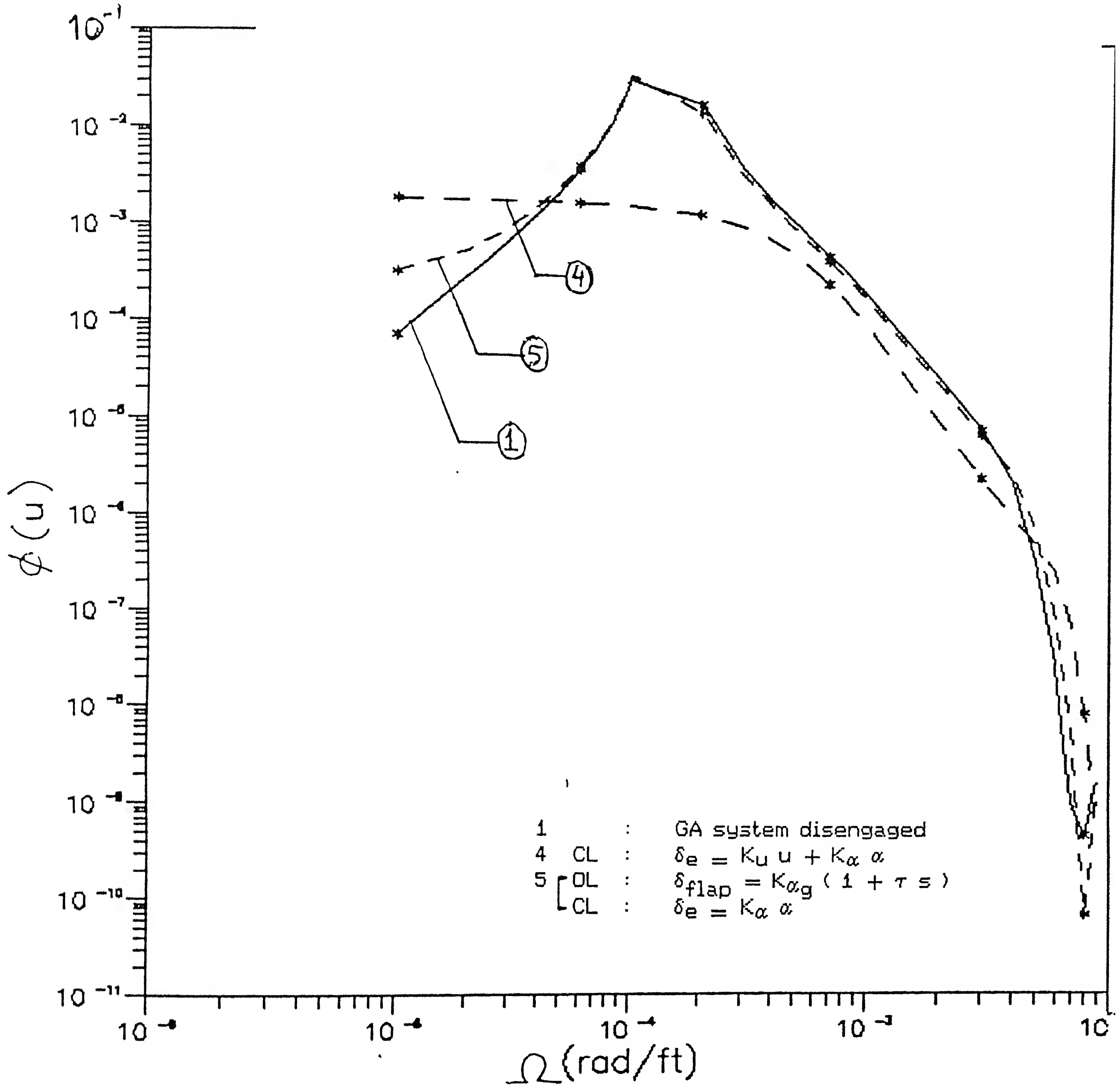


Fig 5.7 Power Spectrum of u w.r.t. Wg for the two promising control laws

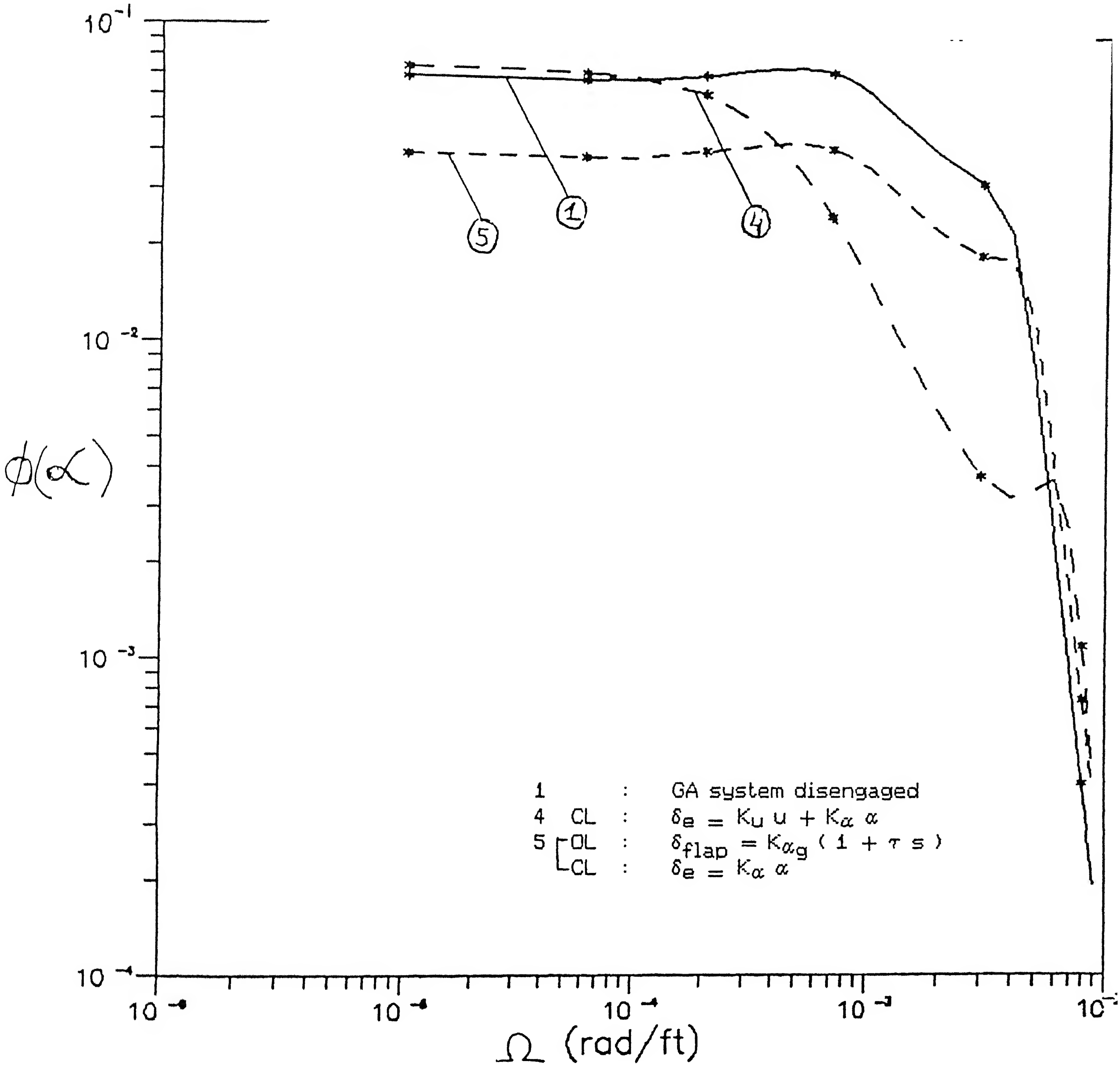


Fig 5.8 Power Spectrum of α w.r.t. Wg for the two promising control laws

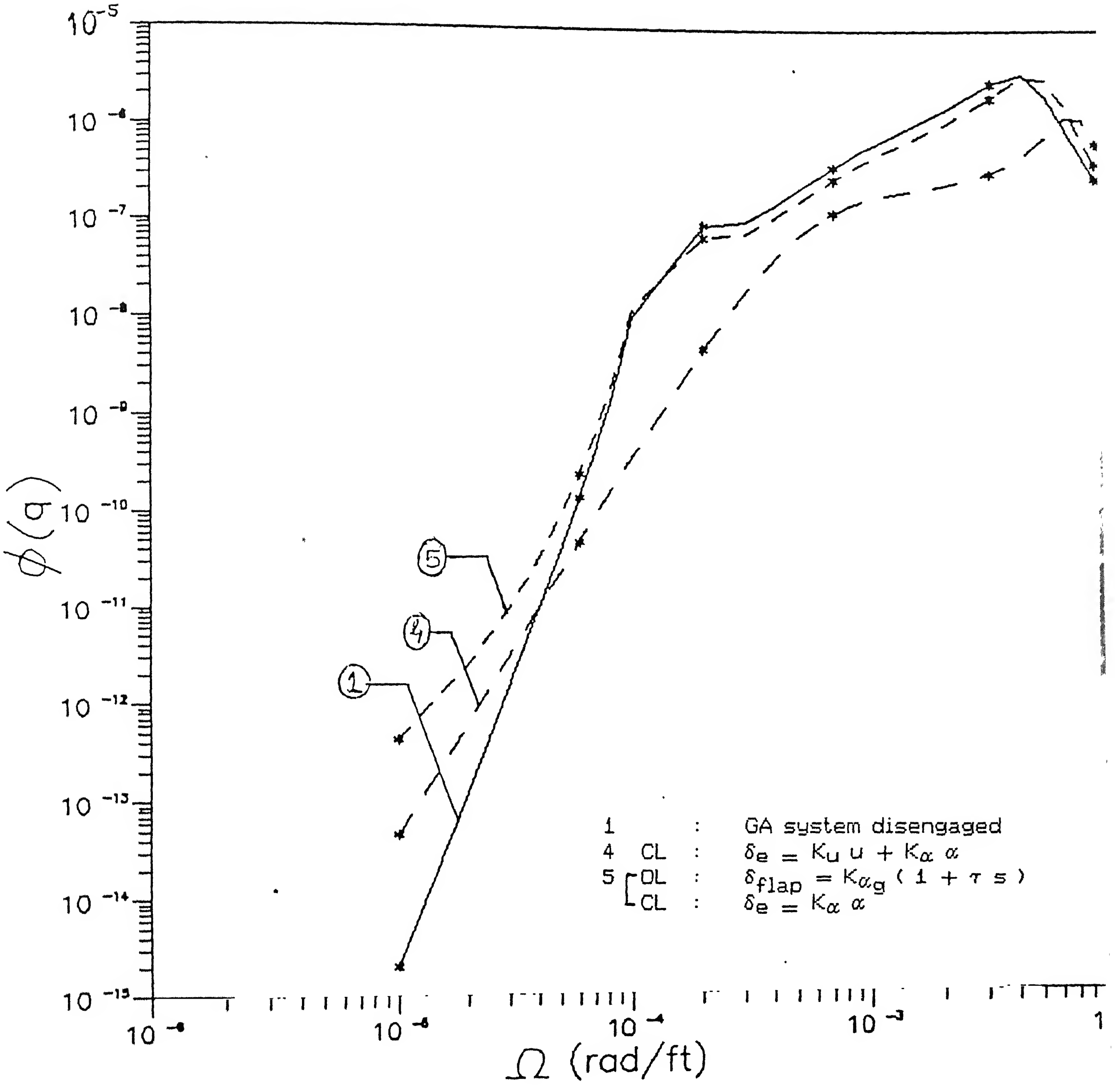


Fig 5.9 Power Spectrum of q w.r.t. Wg for the two promising control laws

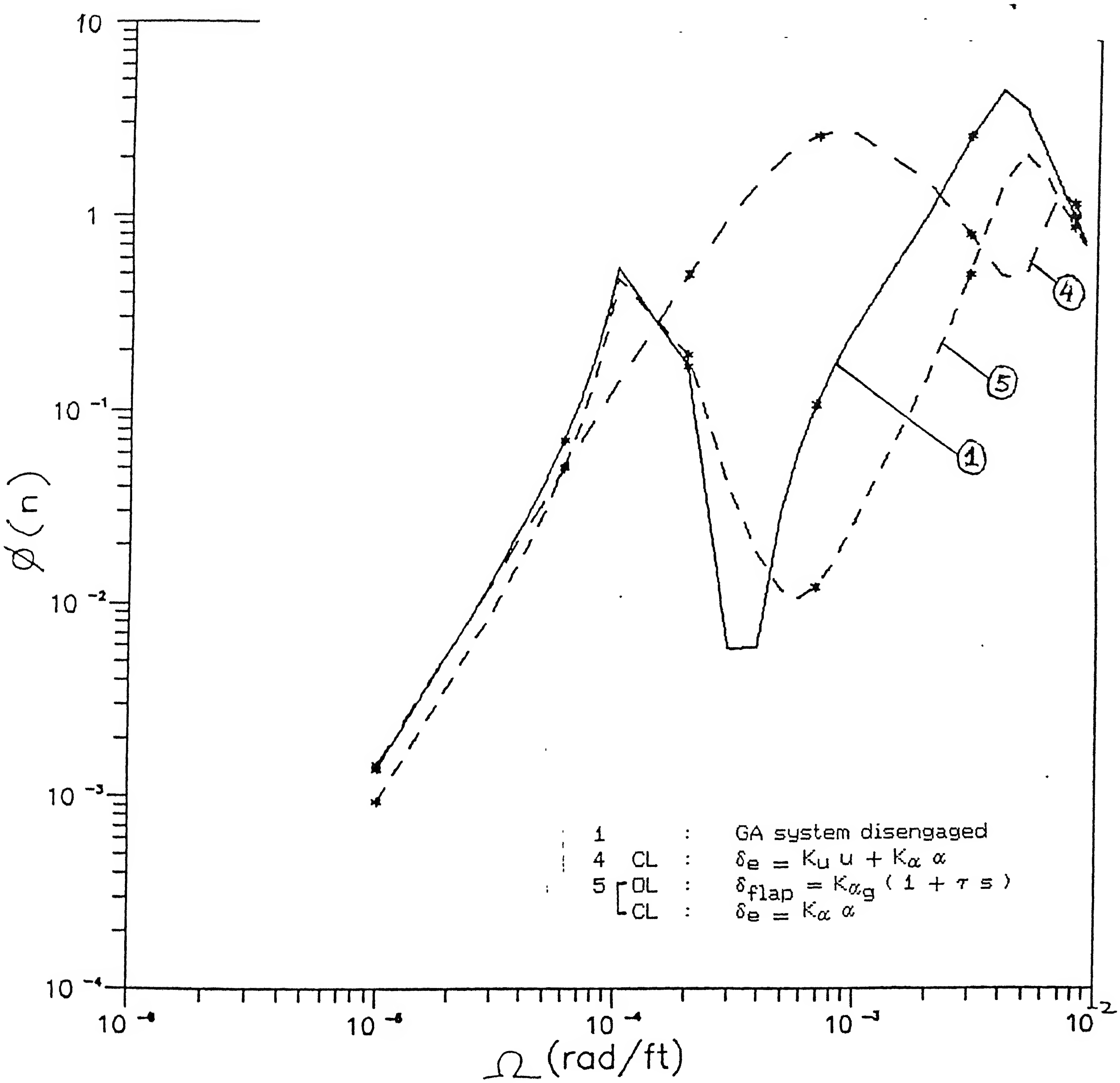


Fig 5.10 Power Spectrum of η w.r.t. Wg for the two promising control laws

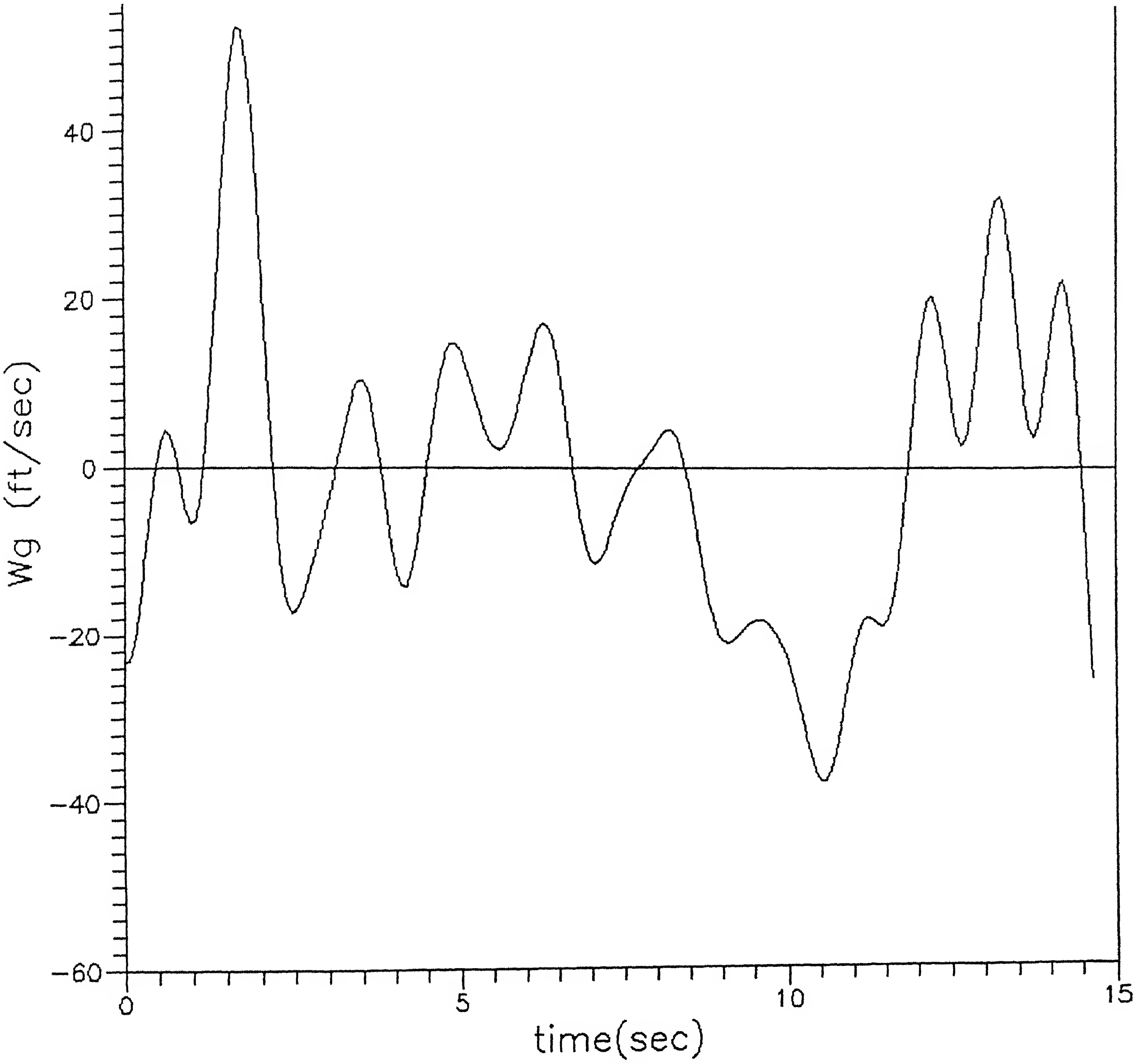


Fig 5.11 Vertical Velocity Wg(t) of the Simulated Random Gust

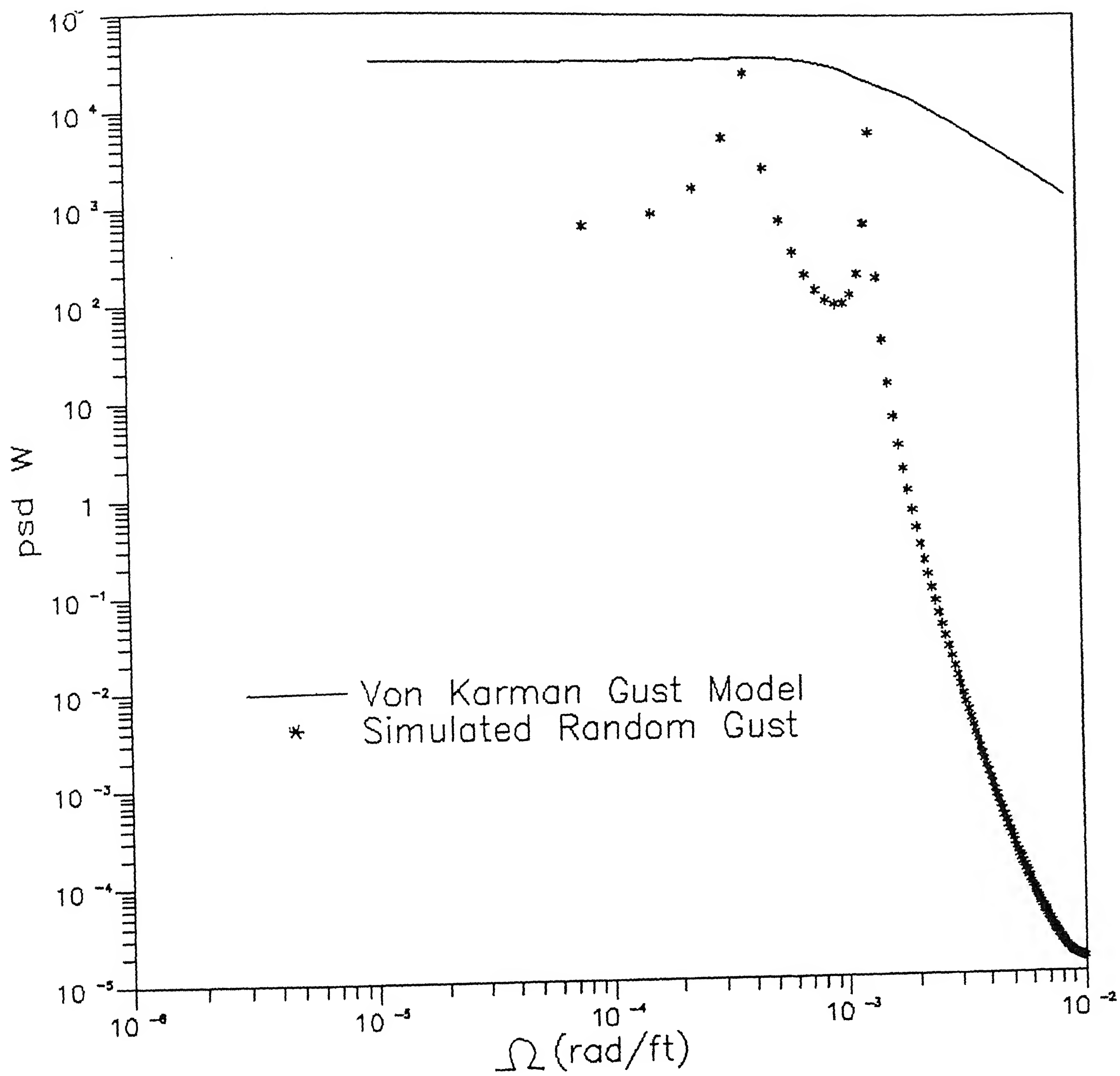


Fig 5.12 Power Spectrum of the Simulated Random Gust

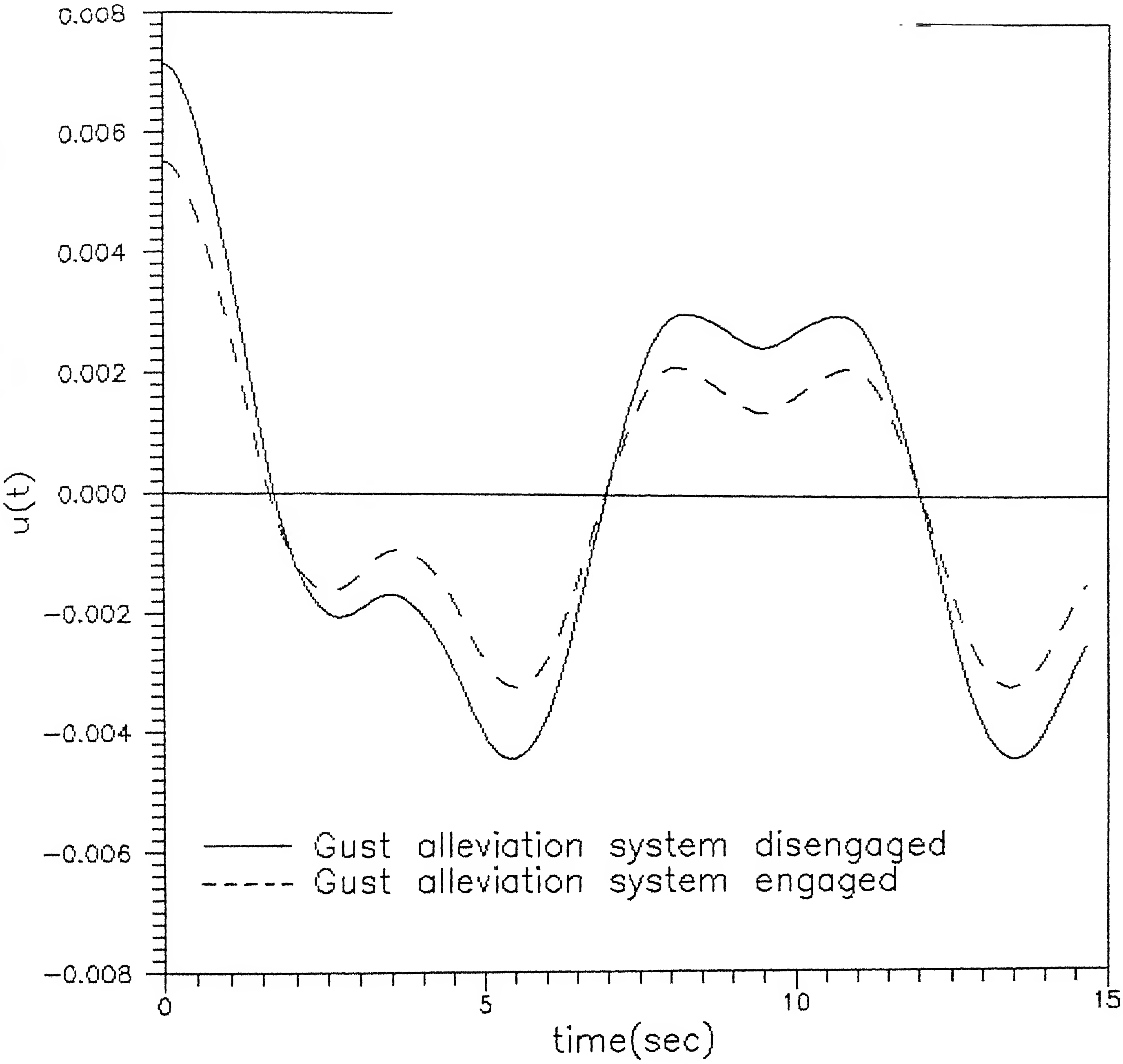


Fig 5.13 Time History of u(t) for the random gust

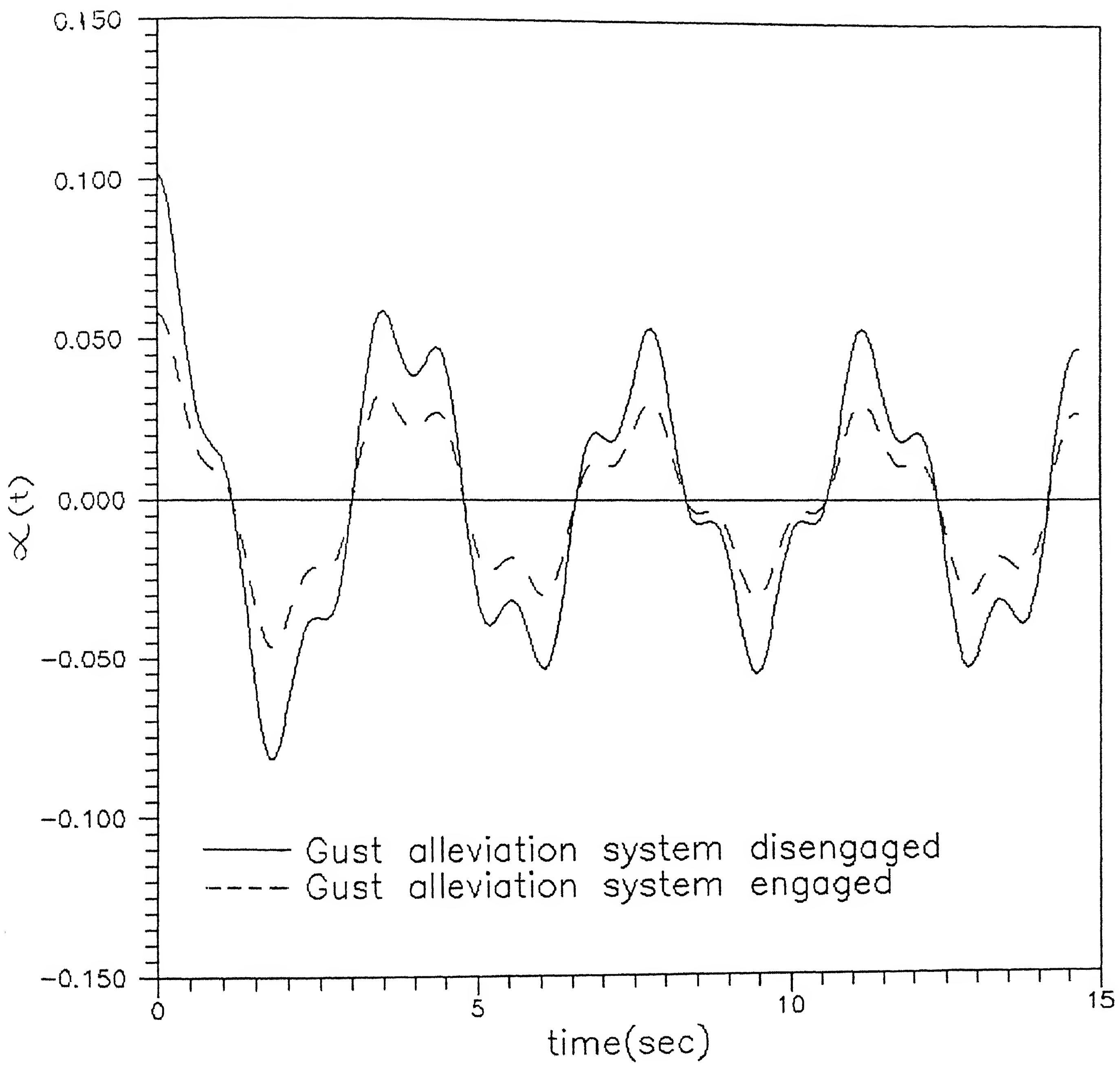


Fig 5.14 Time History of $\alpha(t)$ for the random gust

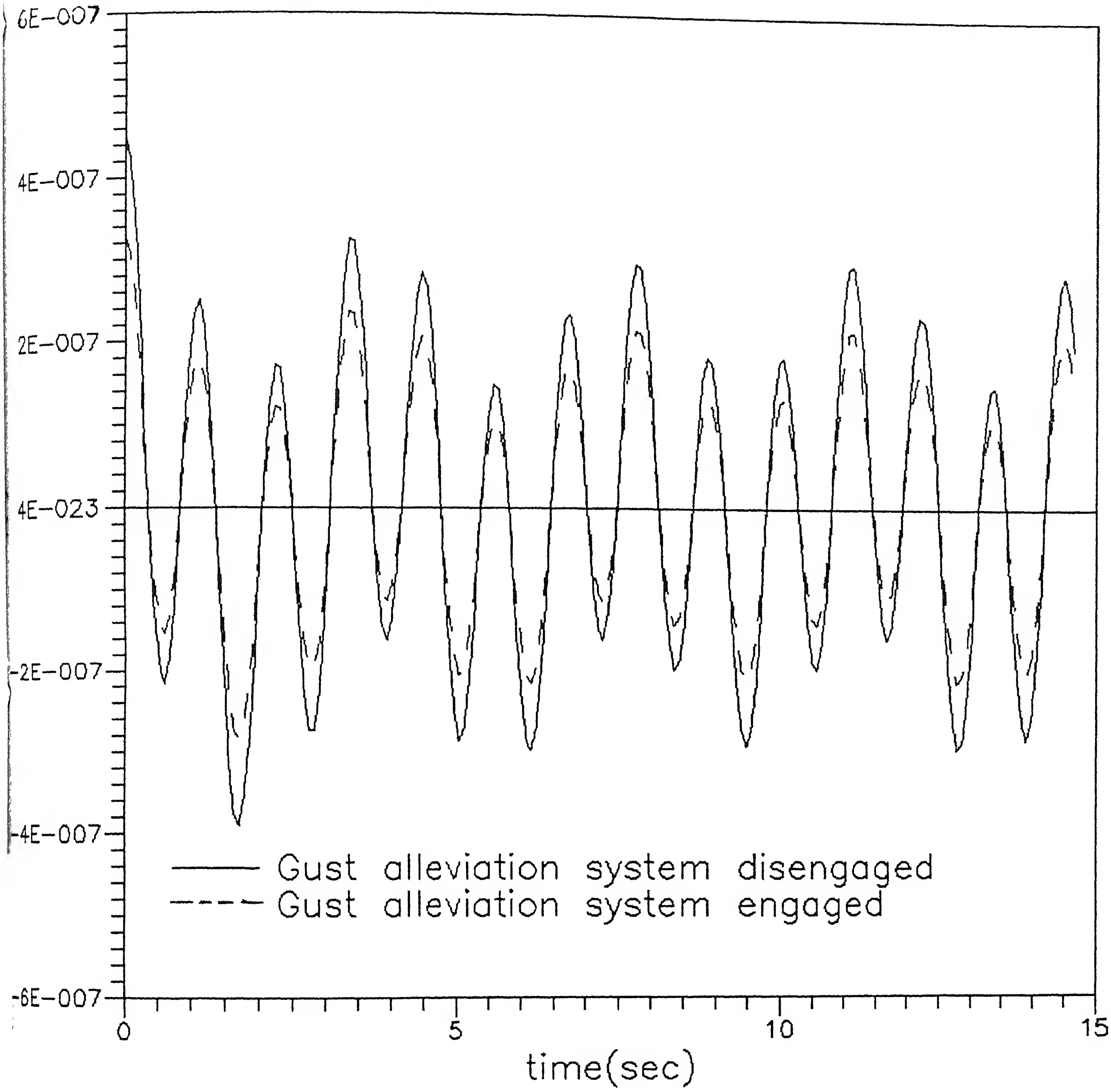


Fig 5.15 Time History of $\hat{q}_2(t)$ for the random gust

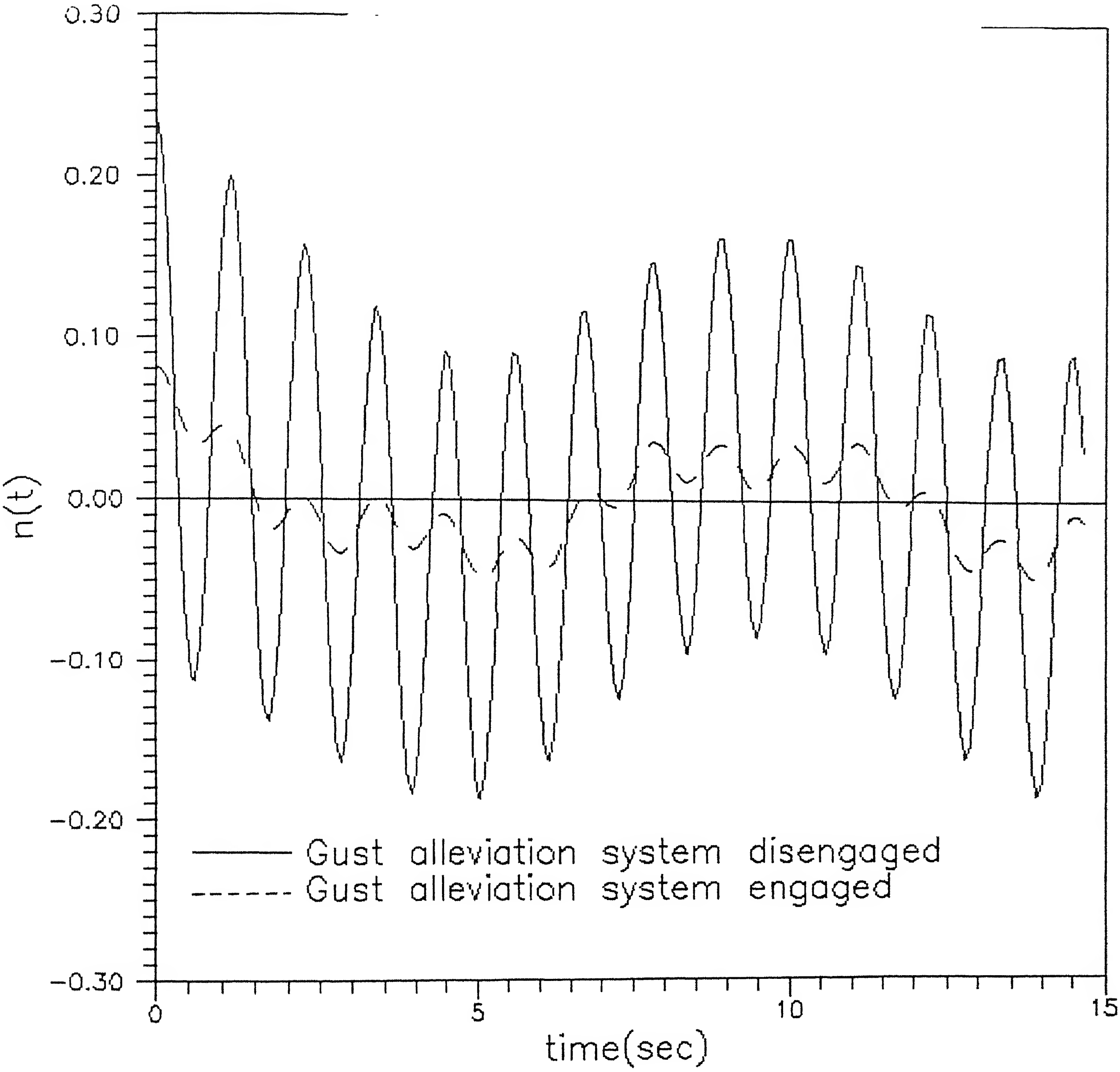


Fig 5.16 Time History of $n(t)$ for the random gust

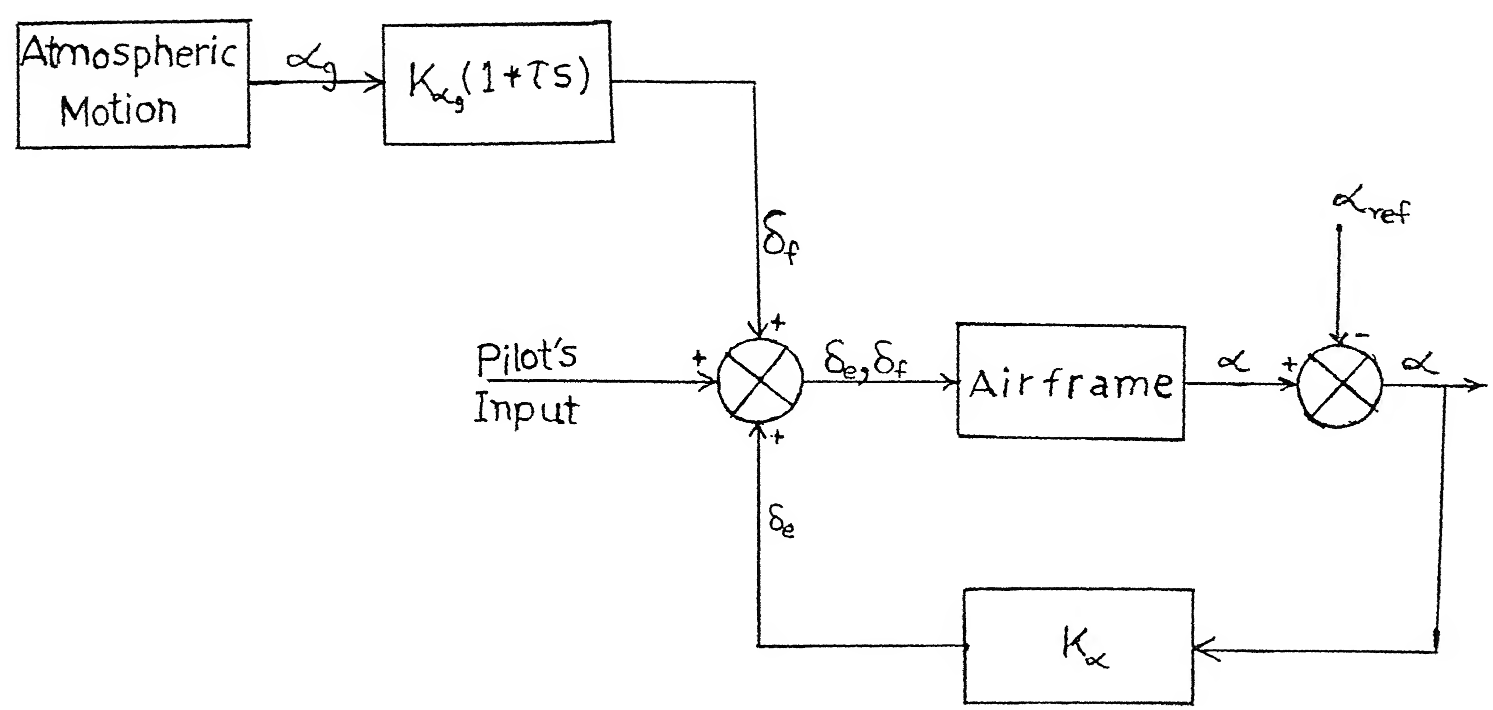


Fig 5·17 Gust Alleviation System

Conclusions

The improvement of passenger comfort is the main aim for gust alleviation of aircraft for small transport airplanes. The efficiency of the gust alleviation system is measured in terms of a simple comfort index adopted by Krag[2]. In view of the above, it is concluded that

(i) A combination of an open loop (incorporating direct measurement of the gust angle of attack and a proportional plus derivative filter to the direct lift control (flaps) device) and a closed loop (incorporating a feedback of the angle of attack α to the elevator) is promising for the gust alleviation of aircraft. The reduction in the comfort index is quite significant, more than other cases without much change in the handling qualities of the aircraft.

(ii) A closed loop (incorporating a feedback of u and angle of attack α to the elevator) also appears promising w.r.t. gust alleviation of aircraft as the reduction in the comfort index is significant but there is a deterioration in the handling qualities of the aircraft due to significant increase in the short period frequency.

(iii) An exploratory study into the effect of relaxation of static stability and its artificial compensation by pitch attitude feedback to the elevator[13] does not help in alleviation of loads and motions that result, when an airplane encounters a gust.

Thus, a combination of feedforward and feedback loop process is helpful for gust alleviation without much change (deterioration) in the handling qualities of the airplane.

Suggestions For Further Improvements

The present work on the study of gust alleviation of aircraft may be extended as below :

1. The study of gust alleviation of aircraft may be extended to both symmetric and asymmetric maneuvres.

2. The effect of time delay which is inherent in complex digital control systems may be accounted for and studied.

3. The study would be more realistic by the inclusion of 3-d turbulence and taking into account, the complete set of the equations of motion.

4. A more comprehensive index of passenger comfort should be used to account for various factors such as noise, location etc.

References

[1]Wilson E.B., Theory of the Airplane Encountering Gusts, NACA Rep, 1915.

[2]Krag B., Active Control Technology for Gust Alleviation, Von Karman Institute for Fluid Dynamics, December 1978.

[3]Philips W.H. and Kraft C.G., Theoritical Study of Some Methods for Increasing Smoothness of Flight Through Rough Air, NACA TN 2416, 1951.

[4]Etkin B., Turbulent Wind And its Effect on Flight, Journal of Aircraft, Vol 18 No. 5, May 1981.

[5]Press H., and Houbolt J.C., Some Applications of Generalized Harmonic Analysis to Gust loads of Airplanes, Journal of Aeronautical Sciences, Vol 22, No. 1, 1955.

[6]Press H. and Mazelsky B., A Study of the Application of Power Spectral Methods of Generalized Harmonic Analysis to Gust Loads on Airplanes, NACA TN 2853, 1953

[7]Diedrich F.W., The Dynamic Response of a Large Airplane to Continuous Random Atmospheric Disturbances, Journal of Aeronautical Sciences, Vol 23 No. 10, 1956.

[8]Eichenbaum F.D., A General Theory of Aircraft Response to 3-d Turbulence, Journal of Aircraft, Vol 8, No. 5, May 1971, pg 353–360.

[9]Liebst B.S., Garrard W.L. and Farm Terome A., Desogn Of Multivariable Flutter Suppression / Gust Load Alleviation System, Journal of Guidance Control and Dynamics, Vol 11 No.3, June 1988.

[10]Houlbolt J.C, Atmospheric Turbulence, AIAA Journal Vol 11 April 1973 pg 421-437

[11]Disney T.E., C-5A Active Load Alleviation System, Journal of Spacecraft and Rockets, Vol 14 February 1977, pg 81-86

[12]Fung W.C., Statistical Aspects of Dynamic Loads, Journal of Aeronautical Sciences, Vol 20 No. 5, May 1953, pg 317

[13]Gopalkrishna S , A Study of the Longitudinal Dynamical Characteristics of Aircraft of Relaxed Static Stability with Pitch Attitude Feedback, M. Tech Thesis, Indian Institute of Tecnology, I.I.T Kanpur, June 1988

[14]Babister A.W., Aircraft Dynamic Stabilty & Response, Pergamon Press, 1980

[15]Etkin B , Dynamics of Atmospheric Flight, New York, London, John Wiley & Sons, Inc., 1965

[16]Clementson Gerhardt C., An Investigation of the Power Spectral Density of Atmospheric Turbulence, PhD. Thesis, MIT, 1950

[17]McFarland Ross A , Human Factors in Air Transport Design, McGraw Hill Book Co., Inc., 1946, pg 367-377

[18]Roshkam J , Flight Dynamics of Rigid and Elastic Airplanes, Lawrence, Kansas, 1972.

[19]Press H., Meadows M. T., A Re-evaluation of Gust Load Statistics for Applications in Spectral Calculations, NACA TN 3540, 1955.

[20]Press W. H., Flannery B. P., Teukolsky S. A. and Vettering W.T , Numerical Recipes : The Art of Scientific Computing, Cambridge University Press, 1986.

[21]Shizouka M , Simulation of Multivariate and Multidimensional Random Processes, Journal of the Accoustical Society of India, Vol. 49, No. 1 (part 2), 1971.

Appendix A

(i) Isotropic Turbulence relations

Some key relations relating to isotropic turbulence :

<u>1-D spectra</u>

$$\phi_u(L\Omega_1) = \int_{L\Omega_1}^{\infty} \frac{E(L\Omega_1)}{L\Omega} \left[1 - \left(\frac{L\Omega_1}{L\Omega}\right)^2 \right] d(L\Omega) \qquad \text{A.1}$$

$$\phi_v(\Omega) = \phi_w(\Omega) = \frac{1}{2}\,\phi_u(\Omega) - \frac{\Omega}{2}\,\frac{d\phi_u}{d\Omega} \qquad \text{A.2}$$

$$\phi_w(\Omega) = \frac{1}{\pi}\int_{-\infty}^{\infty} R_w(x)\, e^{-i\Omega x}\, dx \qquad \text{A.3}$$

<u>1-D correlation function</u>

$$R_w(x) = \frac{1}{X}\int_0^X w(\xi)\, w(x+\xi)\, d\xi \qquad \text{A.4}$$

$$R_w(x) = \int_0^{\infty} \phi_w(\Omega)\cos(\Omega x)\, dx \qquad \text{A.5}$$

$$\sigma_u = \sigma_v = \sigma_w \qquad \text{A.6}$$

$$L = L_u = 2L_w = \frac{2}{\sigma_w^2}\int_0^{\infty} R_w(x)\, dx \qquad \text{A.7}$$

$$\phi_w(\Omega)\Big|_{\Omega=0} = \frac{\sigma_w^2 L}{\pi} \qquad \text{A.8}$$

$$\phi(\omega) = \left[\frac{1}{2\pi}\right]\phi(f) = \left[\frac{1}{V}\right]\phi(\Omega) = \left[\frac{L}{V}\right]\phi(L\Omega) \qquad \text{A.9}$$

$\omega = 2\pi f = V\Omega$ A.10

The 1–D spectrum ϕ_u is defined in terms of a general 3–D spectrum $E(\Omega)$; $\Omega = \frac{\omega}{V}$ where ω = circular frequency and V = speed of flight. The vertical spectrum ϕ_w is expressible in terms of the longitudinal spectrum ϕ_u.

The auto correlation function R_w and the spectrum ϕ_w are the Fourier transform pairs but because they are both symmetrical can be expressed simply by cosine transforms. The area under the spectrum represents the mean square value of turbulence. The initial value of the auto correlation function by definition is also the mean square value. The integral scale L is defined as twice the area under the R_w correlation function normalized by σ_w^2. The initial value of the ϕ_w spectrum (using the frequency argument $\Omega = \frac{\omega}{V}$) is $\sigma_w^2 \frac{L}{\pi}$.

The form adopted by Von Karman for $E(\Omega)$ is as below

$$E(\Omega) = C \frac{(\Omega/\Omega_0)^4}{[\ 1 + (\Omega/\Omega_0)^2]^{17/6}} \qquad \text{A.11}$$

This exhibits a behaviour of Ω^{-4} and $\Omega^{\frac{-5}{3}}$ at low and high frequencies respectively. The constants C and Ω_0 and the one-dimensional spectra that are obtained from this relation by using the above relations are as below :

$$\phi_u = \sigma_u^2 \left(\frac{2L}{\pi}\right) \frac{1}{[\ 1 + (1.339L\Omega)^2]^{5/6}} \qquad \text{A.12}$$

$$\phi_w = \sigma_w^2 \left(\frac{L}{\pi}\right) \frac{[1 + \frac{8}{3}(1.339L\Omega)^2]}{[\ 1 + (1.339L\Omega)^2]^{11/6}} \qquad \text{A.13}$$

$$C = \left(\frac{55}{9}\right) \left(\frac{L}{\pi}\right) \sigma_w^2 \qquad \text{A.14}$$

$$\Omega_0 = \frac{1}{1.339L} \qquad \text{A.15}$$

(ii) Simulation of random gust

[20-21]The random gust is simulated by considering the vertical gust as a Gaussian random process $W_g(t)$ with mean zero and the mean square spectral density function $S_0(\omega)$. This process can be simulated by way of the following series :

$$f(t) = \sigma \sqrt{\frac{2}{N}} \sum_{k=1}^{N} \cos (\omega_k t + \phi_k) \qquad \text{A.16}$$

where

$$\sigma^2 = \int_{-\infty}^{\infty} S_0(\omega) \, d\omega \qquad \text{A.17}$$

σ is defined as the standard deviation of the process $W_g(t)$, ω_k (k = 1, 2, ..., N), are independent random variables identically distributed with the density function $g(\omega) \equiv g(\omega_k)$ obtained by normalizing $S_0(\omega)$;

$$g(\omega) = \frac{S_0(\omega)}{\sigma^2} \; ; \qquad \text{A.18}$$

and ϕ_k are independendent random variables identically distributed with the uniform density $\frac{1}{2\pi}$ between 0 and 2π. Note that ω_k and ϕ_l (k,l = 1, 2, ..., N) are independent.

[13]Gopalkrishna S., A Study of the Longitudinal Dynamical Characteristics of Aircraft of Relaxed Static Stability with Pitch Attitude Feedback, M. Tech Thesis, Indian Institute of Tecnology, I.I.T. Kanpur, June 1988

[14]Babister A.W., Aircraft Dynamic Stabilty & Response, Pergamon Press, 1980

[15]Etkin B., Dynamics of Atmospheric Flight, New York, London, John Wiley & Sons, Inc., 1965

[16]Clementson Gerhardt C., An Investigation of the Power Spectral Density of Atmospheric Turbulence, PhD. Thesis, MIT, 1950

[17]McFarland Ross A., Human Factors in Air Transport Design, McGraw Hill Book Co., Inc., 1946, pg 367-377

[18]Roshkam J., Flight Dynamics of Rigid and Elastic Airplanes, Lawrence, Kansas, 1972.

[19]Press H., Meadows M. T., A Re-evaluation of Gust Load Statistics for Applications in Spectral Calculations, NACA TN 3540, 1955.

[20]Press W. H., Flannery B. P., Teukolsky S. A. and Vettering W.T., Numerical Recipes : The Art of Scientific Computing, Cambridge University Press, 1986.

[21]Shizouka M., Simulation of Multivariate and Multidimensional Random Processes, Journal of the Accoustical Society of India, Vol. 49, No. 1 (part 2), 1971.

Appendix B
(Non Dimensionalization)

The dimensional derivatives used in the equations of motion are defined as below:

Table B.1.

$$X_u = \frac{1}{m}\frac{\partial X}{\partial u} \qquad X_\alpha = \frac{1}{m}\frac{\partial X}{\partial \alpha}$$

$$Z_u = \frac{1}{m}\frac{\partial Z}{\partial u} \qquad Z_\alpha = \frac{1}{m}\frac{\partial Z}{\partial \alpha} \qquad Z_{\dot\alpha} = \frac{1}{m}\frac{\partial Z}{\partial \dot\alpha} \qquad \text{etc.}$$

$$M_u = \frac{1}{I_{yy}}\frac{\partial M}{\partial u} \qquad M_\alpha = \frac{1}{I_{yy}}\frac{\partial M}{\partial \alpha} \qquad M_\eta = \frac{1}{I_{yy}}\frac{\partial M}{\partial \eta} \qquad \text{etc.}$$

$$M_{T_u} = \frac{1}{I_{yy}}\frac{\partial M_T}{\partial u} \qquad M_{T_\alpha} = \frac{1}{I_{yy}}\frac{\partial M_T}{\partial \alpha}$$

Non dimensionalized stability derivative relations

The various non dimensional systems in use vary in :

(i) The definition of the time constant t^*

(ii) The choice of the characteristic length l

(iii) The definitions of the non dimensional stability derivatives

The system followed throughout the present work uses the NACA stability derivatives with time constant $t^* = \frac{1}{U_0}$ and the characteristic length used in the longitudinal equations of motion as $l = \frac{\bar{c}}{2}$. The relative mass or density parameter μ defined as the ratio of the airplane mass to the volume Sl of air is expressed as $\mu = \frac{m}{\rho S l}$.

The non dimensional system used is defined in Table B.2 below :

Table B.2.

Dimensional Quantity	Divisor	Non Dimensional Quantity
X, Z	$\frac{1}{2}\rho u_0^2 S$	C_X , C_Z
L, M	$\rho u_0^2 S l$	C_L , C_m
u, w	U_0	$\hat{u}$, α
q	$\frac{1}{t^*}$	$\hat{q}$
θ	—	θ
I_{yy}	$\rho S l^3$	i_B
t	t^*	$\hat{t}$
$\frac{d}{dt}$	$\frac{1}{t^*}$	$D = \frac{d}{d\hat{t}}$

The non dimensional derivatives are related to the stability derivatives given for the aircraft as in Table B.3 below :

Table B.3.

$C_X = C_{T_e} - C_{D_e}$	$C_Z = -C_{L_e}$	
$C_{X_u} = C_{T_u} - C_{D_u}$	$C_{Z_u} = -C_{L_u}$	$C_{m_u} = C_{m_u}$
$C_{X_\alpha} = C_{L_e} - C_{D_\alpha}$	$C_{Z_\alpha} = -(C_{D_e} + C_{L_\alpha})$	$C_{m_\alpha} = C_{m_\alpha}$
—	$C_{Z_{\dot{\alpha}}} = -C_{L_{\dot{\alpha}}}$	$C_{m_{\dot{\alpha}}} = C_{m_{\dot{\alpha}}}$
$C_{X_q} = -C_{D_q}$	$C_{Z_q} = -C_{L_q}$	$C_{m_q} = C_{m_q}$
$C_{X_\eta} = -C_{D_\eta}$	$C_{Z_\eta} = -C_{L_\eta}$	$C_{m_\eta} = C_{m_\eta}$

The nomenclature used in the equations of motion after incorporating the suitable modifications and used in the computer program is as below :

Table B.4.

$$T_f = \frac{2u_0^2}{g\bar{c}}$$

$A = 2\mu$

$B = -Cx_u - 2C_{L_e}\tan\theta_e - Cx_\eta K_u$

$C = Cx_\eta K_n T_f$

$D = -Cx_\alpha - Cx_\eta K_\alpha$

$E = -Cx_\eta (K_q + K_n T_f)$

$F = C_{L_e} - Cx_\eta K_\theta$

$G = 2C_{L_e} - Cz_u - Cz_\eta K_u$

$H = 2\mu - Cz_{\dot{\alpha}} + Cz_\eta K_n T_f$

$I = -Cz_\alpha - Cz_\eta K_\alpha$

$J = -(2\mu + Cz_q) - Cz_\eta (K_q + K_n T_f)$

$K = -C_{L_e}\tan\theta_e - Cz_\eta K_\theta$

$L = -Cm_u - Cm_\eta K_u$

$M = -Cm_{\dot{\alpha}} + Cm_\eta K_n T_f$

$O = -Cm_\alpha - Cm_\eta K_\alpha$

$P = i_B$

$Q = -Cm_q - Cm_\eta (K_q + K_n T_f)$

$R = -Cm_\theta - Cm_\eta K_\theta$

$U = -Cx_\alpha + Cx_\delta K_{\alpha_g}$

$V = Cz_q - Cz_{\dot{\alpha}} + Cz_\delta K_{\alpha_g}\tau_{filter}$

$W = -Cz_\alpha + Cz_\delta K_{\alpha_g}$

$X = Cm_q - Cm_{\dot{\alpha}} + Cm_\delta K_{\alpha_g}\tau_{filter}$

$Y = -Cm_\alpha + Cm_\delta K_{\alpha_g}$

$T = Cx_\delta K_{\alpha_g}\tau_{filter}$

$E1 = Cx_\eta$

$E2 = Cz_\eta$

$E3 = Cm_\eta$

Appendix C

(Harmonic Analysis Theory)

The theory of harmonic analysis indicates that an arbitrary periodic function can be represented by a Fourier series. In order to apply this technique to non-periodic phenomena, the Fourier series takes the form of the Fourier integral. A neccessary condition for the frequency representation $G(\omega)$ of a time function $F(t)$ to exist, is that the Fourier integrals must be convergent. In order to develop a frequency representation applicable to continuous disturbances, the theory of random processes makes use of the concept of a stationary random process ie. the disturbance is invariant with time and a statistical equilibrium exists. For a stationary random function $y(t)$, the mean square $\overline{y^2(t)}$ represents a measure of the disturbance intensity.

$$\overline{y^2(t)} = \lim_{T\to\infty} \frac{1}{T} \int_0^T [y(t)]^2 \, dt \qquad \text{C.1}$$

The function $y(t)$ is considered to be composed of an infinite number of sinusoidal components with circular frequency ω, between 0 and ∞. The portion of $\overline{y^2(t)}$ arising from the components having frequencies between ω and $\omega+d\omega$ is denoted by $\phi(\omega)d\omega$. The function $\phi(\omega)$ is called the power spectral density (p. s. d.) function. Atmospheric turbulence is a stationary random process that can be statistically described by a single reduced p.s.d. curve.

$$\overline{y^2(t)} = \int_0^\infty \phi(\omega)d\omega \qquad \text{C.2}$$

$$\phi(\omega) = \lim_{T\to\infty} \frac{1}{\pi} \left| \int_0^T y(t)\, e^{-i\omega t} \, dt \right|^2 \qquad \text{C.3}$$

where | | indicates the modulus of the complex quantity. Thus $\phi(\omega)d\omega$ is a measure of the contribution of that frequency to $\overline{y^2(t)}$. The system frequency response $T(i\omega)$ is defined such that $T(i\omega)e^{i\omega t}$ is the system response to the sinusoidal input $e^{i\omega t}$.

$$T(i\omega) = i\omega \int_0^{\infty} A(t)\, e^{-i\omega t}\, dt \qquad \text{C.4}$$

where $A(t)$ is the response to unit step. The p.s.d. of $\hat{\phi}_i(\Omega)$ may be found by first determining the airplane response to a step gust and then using this step response in equation (8) above to determine the frequency response for a continuous sinusoidal gust input.

The auto correlation function $R(\tau)$ is defined as

$$R(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_0^{\infty} y(t)\, y(t+\tau)\, dt \qquad \text{C.5}$$

and it is reciprocally related to the p.s.d. function as

$$R(\tau) = \int_0^{\infty} \phi(\omega) \cos(\omega\tau)\, d\omega \qquad \text{C.6}$$

$$\phi(\omega) = \frac{2}{\pi} \int_0^{\infty} R(\tau) \cos(\omega\tau)\, d\tau \qquad \text{C.7}$$

A change of variables is appropriate in the gust case because in these terms power spectra are independent of the airplane forward speed. Use is made of the distance x in feet and the associated frequency arguments Ω. Thus the turbulent vertical velocity distribution along a line in space can, at an instant of time, be considered to represent a stationary random function of space consisting of an infinite number of harmonics of various frequencies or wavelengths.

$$\Omega=\frac{\omega}{V} \qquad ; \qquad x = Vt \tag{C.8}$$

The variable Ω is the reduced frequency in radians per foot and variable x is the airplane flight distance in feet. Thus,

$$\hat{\phi}(\Omega) = V\phi(\omega) = V\phi(V\Omega) \tag{C.9}$$

The resulting power spectrum of atmospheric turbulence is independent of the airplane speed and in a limited sense may represent a universal reduced power spectral density function of atmospheric turbulence.

$$\hat{T}(i\Omega) = i\Omega \int_0^\infty \hat{A}(x)\, e^{-i\Omega x}\, dx \tag{C.10}$$

where $\hat{A}(x) = A(t)$ = response to unit step gust input

The input output relation in terms of Ω ca be written as

$$\hat{\phi}_O(\Omega) = \hat{\phi}_i(\Omega) \quad | \; T(i\Omega) \; |^2 \tag{C.11}$$

The probability distribution of the output can be represented by a normal distribution with mean 0 and the standard deviation σ obtained as below

$$\sigma^2 = \int_0^\infty \phi_O(\omega)\, d\omega = \int_0^\infty \hat{\phi}_O(\Omega)\, d\Omega \tag{C.12}$$

This relation ties the normally distributed loads to the time history and is thus of importance in the gust load analysis

Appendix D

(Longitudinal Derivatives of Example Aircraft)

Longitudinal Aerodynamic Characteristics of Business Jet Transport Aircraft

$W = 13000$ lbs	$h = 40000$ ft	$\rho = 0.000588$ slug ft^{-3}
$I_{yy} = 18800$ slug ft^2	$M = 0.7$	$\bar{c} = 7.04$ ft
$\bar{X}_{cg} = 0.315$	$S = 232$ ft^2	$\Theta_e = 0$ (stability axes)
$\alpha_e = 2.7^0$		

Non Dimensional Derivative	Cruise $U_e = 675$ ftsec^{-1}	Non Dimensional Derivative	Cruise $U_e = 675$ ft sec^{-1}
C_{D_u}	0.0	$C_{L_{\delta_e}}$	0.556
C_{D_e}	0.033	C_{m_u}	0.05
$C_{T_{x_u}}$	0.0	C_{m_e}	0.007
$C_{T_{x_e}}$	0.033	$C_{m_{T_u}}$	0.0034
C_{D_α}	0.30	$C_{m_{T_e}}$	0.007
C_{L_e}	0.410	C_{m_α}	0.64
$C_{D_{\delta_e}}$	0.0	C_{m_T}	0.0
C_{L_u}	0.40	$C_{m_{\dot\alpha}}$	6.7
C_{L_α}	5.84	C_{m_q}	15.50
$C_{L_{\dot\alpha}}$	2.20	$C_{m_{\delta_e}}$	1.52
C_{L_q}	4.70		

For the flap used : $C_{D_{\delta f}} = 0$ $\quad C_{L_{\delta f}} = C_{L_e}$ $\quad C_{m_{\delta f}} = 0$

Appendix E
(Program Listing)

```
rogram thesis(input,output);
const
   elevator_deflection=0.0;
   delta_time=0.074;
   Initial_time=0.0;
   Final_time=9.0;
   Initial_Altitude=40000.0;
   Initial_omega=1E-5;                nn=256;
   Final_omega=1E-2;                  nnby2=128;
   gr=32.2;                           nn2=512;
   pi=3.1415926;                      Order_m=4;
   np=7;                              Isign=-1;
   mp=8;
   ftol=1E-6;
   nnk=3;
   eps=1E-6;
   Total_N=256;
type
   perturbation_quantity = (u,w,q,theta);
   matrix        = array [perturbation_quantity] of real;
   PlaneInfoType=(Ve,Rho,
                  S,c,Iyy,thetaE,Muu1,
                  CLe,CLu,CLalpha,CLalphaDot,CLq,CLdeltaE,
                  CDe,Cte,CDu,CTu,CDalpha,CDalphaDot,CDq,CDdeltaE,
                  CMu,CMalpha,CMalphaDot,CMq,CMtheta,CMdeltaE);
  DerArr        = array[PlaneInfoType] of real;
  glmpnp        = ARRAY [1..mp,1..np] OF real;
  glmp          = ARRAY [1..mp] OF real;
  glnp          = ARRAY [1..np] OF real;
  double        = real;
  galr          = array[1..97] of real;      gldarray     = array[1..nn2] of real;
```

```
  glnarray      = array[1..Total_N] of real;
  glmarray      = array[1..Order_m] of real;
var
  PlaneProg                           :DerArr;
  CXu,CX,CXalpha,CZu,CZ,CZalpha,CXq,Czq   :real;
  CZalphaDot,CXalphaDot,CXdeltaE,CZdeltaE:real;
  CXdeltaFlap,CZdeltaFlap             :real;
  CMdeltaFlap                         real;
  DskF1,p11,f11,k11,inp               :text;
  kk1,kk2,kk3,kk4,kk5                 :text;
  psd_w,big_omega                     :real;
  Frequency,dim_frequency             :real;
  Sigma_w,Scale_turbulence            :real;
  response                            :matrix;
  time                                :real;
  FWg,Wg,Wmax                         :real;
  Load_factor                         :real;
  Ph_u_gust,Ph_w_gust,Ph_q_gust  :real;
  Ph_theta_gust,Ph_LF_gust            :real;
  Ph_u_elev,Ph_w_elev,Ph_q_elev  :real;
  Ph_theta_elev,Ph_LF_elev            :real;
  TF_u_gust,TF_w_gust,TF_q_gust  :real;
  TF_theta_gust,TF_LF_gust            :real;
  TF_u_elev,TF_w_elev,TF_q_elev  :real;
  TF_theta_elev,TF_LF_elev            :real;
  Out_w_gust,Out_q_gust               :real;
  Out_LF_gust                         :real;
  Out_u_gust,Out_theta_gust           :real;
  ACMu,ACMalpha,ACMq,ACMdeltaE   :real;
  ACMalphadot,Muu,t_star,iB           :real;
  ACMtheta                            :real;
  Ch_length,ACLe,CLetanthetaE    :real;
  Ku,Kalpha,Kq,Ktheta,KLdFctr    :real;
  Kalpha_gust,T_Filter                :real;
  Aa,Ab,Ac,Ad,Ae,Af,Ag,Ah             :real;
  Ai,Aj,Ak,Al,Am,Ao,Ap,Aq             :real;
```

```
    Ar,Gt,Gu,Gv,Gw,Gx,Gy              :real;
    E1,E2,E3                          :real;
    TFactor,KFactor_CLaw              :real;
    sum1,sum2,comfort_criterion       :real;
    delta_omega                       :real;
    check_coeff,count                 :real;
    temporarily                       :real;
    i,iter,j,ndim                     :integer;
    x                                 :glnp;
    y                                 :glmp;
    p                                 :glmpnp;
    pr                                :glnp;
    Omegak,Phik,Time_Data             :glnarray;
    random_Phi,random_omega           :real;
    random_Time_Func                  :real;
    accept                            :boolean;
    lomega,Iphi,Irandom_y             :integer;
    iicount,nkk1                      :integer;
    gomega,gphi,grandom_y             :galr;
    golix1,golix2,golix3              :integer;
    gplix1,gplix2,gplix3              :integer;
    gylix1,gylix2,gylix3              :integer;      Moutu,Moutw,Moutq,MoutLF     :gldarray;
    wk1,wk2                           :glnarray;      Mouttheta                   :gldarray;
    wkm                               :glmarray;       Frequencys                 :real;
    fdt,pm                            :real;           temp                       :integer;
    cof                               :glmarray;
    nnk1,kkm                          :integer;

FUNCTION sngl(xx:real):real;
 BEGIN
  sngl := xx
 END;

Procedure ReadDskForAcDer;
 var
   i            :PlaneInfoType;
```

```
Procedure ReadDskForAcDer;
 var
   i          :PlaneInfoType;
 begin   { ReadDskForAcDer }
    for i:=Ve to CMdeltaE do
     begin
       Readln(DskF1,PlaneProg[i]);
     end;
 end;    { ReadDskForAcDer }

procedure Am_ConstCoeff;
 begin { Am_ConstCoeff}
    CXu:=PlaneProg[CTu]-PlaneProg[CDu];
    CX:=PlaneProg[CTe]-PlaneProg[CDe];
    CXalpha:=PlaneProg[CLe]-PlaneProg[CDalpha];
    CZu:=-PlaneProg[CLu];
    CZ:=-PlaneProg[CLe];
    CZalpha:=-(PlaneProg[CDe]+PlaneProg[CLalpha]);
    ACMu:=PlaneProg[CMu];
    ACMalpha:=PlaneProg[CMalpha];
    CXq:=-PlaneProg[CDq];
    CZq:=-PlaneProg[CLq];
    ACMq:=PlaneProg[CMq];
    CZalphaDot:=-PlaneProg[CLalphaDot];
    CXdeltaE:=-PlaneProg[CDdeltaE];
    CZdeltaE:=-PlaneProg[CLdeltaE];
    ACMdeltaE:=PlaneProg[CMdeltaE];
    CXdeltaFlap:=0.0;
    CZdeltaFlap:=-0.2*PlaneProg[Cle];
    CMdeltaFlap:=0.0;
    CXalphaDot:=-PlaneProg[CDalphaDot];
    ACMalphadot:=PlaneProg[CMalphaDot];
    ACMtheta:=PlaneProg[CMtheta];
    Muu:=PlaneProg[Muu1];
    t_star:=PlaneProg[c]/(2*PlaneProg[Ve]);
    Ch_length:=PlaneProg[c]/2;
```

```
    iB:=PlaneProg[Iyy]/(PlaneProg[Rho]*PlaneProg[S]*
 sqr(Ch_length)*Ch_length);
    ACLe:=PlaneProg[CLe];
    CLetanthetaE:=PlaneProg[CLe]*sin(PlaneProg[thetaE])/
cos(PlaneProg[thetaE]);
 end;  { Am_ConstCoeff }

procedure Dummy_Coeffs;
 begin { Dummy_Coeffs }
  TFactor:=(2*sqr(PlaneProg[Ve]))/(gr*PlaneProg[c]);
  Aa:=2*Muu;
  Ab:=-CXu-2*CLetanthetaE-CXdeltaE*Ku;
  Ac:=CXdeltaE*TFactor*KLdFctr;
  Ad:=-CXalpha-CXdeltaE*Kalpha;
  Ae:=-CXdeltaE*(Kq+TFactor*KLdFctr);
  Af:=ACLe-CXdeltaE*Ktheta;
  Ag:=2*ACle-CZu-CZdeltaE*Ku;
  Ah:=2*Muu-CZalphadot+CZdeltaE*TFactor*KLdFctr;
  Ai:=-CZalpha-CZdeltaE*Kalpha;
  Aj:=-(2*Muu+CZq)-CZdeltaE*(Kq+TFactor*KLdFctr);
  Ak:=-CLetanthetaE-CZdeltaE*Ktheta;
  Al:=-ACMu-ACMdeltaE*Ku;
  Am:=-ACMalphadot+ACMdeltaE*TFactor*KLdFctr;
  Ao:=-ACMalpha-ACMdeltaE*Kalpha;
  Ap:=iB;
  Aq:=-ACMq-ACMdeltaE*(Kq+TFactor*KLdFctr);
  Ar:=-ACMtheta-ACMdeltaE*Ktheta;
  Gt:=CXdeltaFlap*T_Filter*Kalpha_gust;
  Gu:=-CXalpha+CXdeltaFlap*Kalpha_gust;
  Gv:=CZq-CZalphadot+CZdeltaFlap*T_Filter*Kalpha_gust;
  Gw:=-CZalpha+CZdeltaFlap*Kalpha_gust;
  Gx:=ACMq-ACMalphadot+CMdeltaFlap*T_Filter*Kalpha_gust;
  Gy:=-ACMalpha+CMdeltaFlap*Kalpha_gust;
  E1:=CXdeltaE;
  E2:=CZdeltaE;
  E3:=ACMdeltaE;
```

```
 end; { Dummy_Coeffs }


procedure Ferrari(A,b,c,d,e:real;
 var re1,re2,re3,re4,im1,im2,im3,im4:real);
 var
  p,q,R,S,smr,sms,mu:real;
Procedure Quad(a,b:real; var Re,Im,Re1,Im1:real);
 var
  D:real;
 begin { Quad }
  D:=a*a-4*b;
  if D<0 then
   begin
    D:=sqrt(-D);
    Re:=-a/2;
    Re1:=Re;
    Im:=D/2 ;
    Im1:=-Im;
   end
  else
   begin
    D:=sqrt(D);
    Re:=( -a+D)/2;
    Re1:=( -a-D)/2;
    Im:=0;
    Im1:=0;
   end;
 end; { Quad }
Procedure NR(var x:real);
Function f(x:real):real;
 begin
  f:=x*x*x-2*c*sqr(x)+S*x-R;
 end;
Function fprime( var x:real):real;
 begin
  fprime:=3*x*x-4*c*x+S;
```

```
 end;
 begin
  repeat
   x:=x-F(x)/Fprime(x);
   writeln('inside NR');
  until abs(F(x)/Fprime(x))<0.0000001;
 end;
Procedure Bab( var mu:real );
 var
  v,OldMu:real;
 begin
  repeat
   writeln('inside Bab');
   OldMu:=mu;
   v:=S+OldMu*(OldMu-2*C);
   Mu:=R/v;
  until abs( mu-OldMu )< 0.0000001 ;
 end;
procedure Birge_Vieta(var x:real);
 var
  aa1      :array[1..nnk] of real;
  bb1,cc1  :array[0..nnk] of real;
  kk       :integer;
 begin { Birge_Vieta }
  aa1[1]:=-2*c;
  aa1[2]:=S;
  aa1[3]:=-R;
  while (bb1[nnk] > eps) do
   begin
    writeln('inside Birge_Vieta');
    bb1[0]:=1;
    cc1[0]:=1;
    for kk:=1 to nnk do
     begin
      bb1[kk]:=aa1[kk]+x*bb1[kk-1];
      cc1[kk]:=bb1[kk]+x*cc1[kk-1];
```

```
      end;
     x:=x-bb1[nnk]/cc1[nnk-1];
    end;
  end; { Birge_Vieta }
  begin  { Ferrari }
   S:=b*d+c*c-4*a*e;
   R:=B*C*D-B*B*E-A*D*D;
   mu:=R/S;
   {NR(mu);
   Bab(mu);}
   Birge_Vieta(mu);
   if not( mu< sqr(b)/(4*a) ) then
    writeln(f11,'The roots of ch eqn do not exist')
   else
    begin
     P:=( (B/A) +sqrt( sqr(b/a)-4*mu/a) )*0.5;
     q:=(b/a)-p;
     smr:=( p*((c/a)-p*q)-(d/a) )/(p-q);
     sms:=c/a - p*q-smr;
     Quad(p,smr,re1,im1,re2,im2);
     Quad(q,sms,re3,im3,re4,im4);
    end;
 end; { Ferrari }

procedure New_ChEqn(var A1,B1,C1,D1,E1:real);
 begin { New_ChEqn }
  A1:=Aa*Ap*Ah;
  B1:=Ab*Ap*Ah+Aa*(Ah*Aq+Ap*Ai-Am*Aj)-Ag*Ac*Ap;
  C1:=Ab*(Ah*Aq+Ap*Ai-Am*Aj)+Aa*(Ah*Ar+Ai*Aq-Am*
      Ak-Ao*Aj)-Ag*(Ac*Aq+Ad*Ap-Am*Ae)+Al*(Ac*Aj-
      Ah*Ae);
  D1:=Ab*(Ah*Ar+Ai*Aq-Am*Ak-Ao*Aj)+Aa*(Ai*Ar-Ao*
      Ak)-Ag*(Ac*Ar+Ad*Aq-Am*Af-Ao*Ae)+Al*(Ac*Ak+
      Ad*Aj-Ah*Af-Ae*Ai);
  E1:=Ab*(Ai*Ar-Ao*Ak)-Ag*(Ad*Ar-Ao*Af)+Al*(Ad*Ak-
      Ai*Af);
```

```pascal
   writeln(f11,'The characteristic eqn is: As4+Bs3+Cs2+Ds+E=0');
   writeln(f11,'A = ',A1);
   writeln(f11,'B = ',B1);
   writeln(f11,'C = ',C1);
   writeln(f11,'D = ',D1);
   writeln(f11,'E = ',E1);
   writeln('The characteristic eqn is: As4+Bs3+Cs2+Ds+E=0');
   writeln('A = ',A1);
   writeln('B = ',B1);
   writeln('C = ',C1);
   writeln('D = ',D1);
   writeln('E = ',E1);
  end; { New_ChEqn }

 procedure Roots;
  var
   R1,R2,I1,I2,R3,I3,R4,I4:real;
   A1,B1,C1,D1,E1:real;
 procedure Real_roots(R11:real);
  var
   t_half,t_double          :real;
  begin { Real_Roots }
   if (R11 < 0.0) then
    begin
     t_half:=(0.69*t_star)/abs(R11);
     writeln(f11,'t_half = ',t_half:7:3,' sec');
     writeln('t_half. = ',t_half:7:3,' sec');
    end
   else
    begin
     t_double:=(0.69*t_star)/abs(R11);
     writeln(f11,'t_double = ',t_double:7:3,' sec');
     writeln('t_double = ',t_double:7:3,' sec');
    end;
 end; { Real_Roots }
procedure Find_Frequency(R11,R22,I11,I22:real);
```

```
var
 Period_T,t_half,N_half      :real;
 t_double,N_double           :real;
begin { Find_Frequency }
 if (I11 <> 0.0) then
  begin
   Period_T:=(2*pi*t_star)/abs(I11);
   writeln(f11,'Period_T = ',Period_T:7:3,' sec');
   writeln('Period_T = ',Period_T:7:3,' sec');
   if (R11 < 0.0) then
    begin
     t_half:=(0.69*t_star)/abs(R11);
     N_half:=t_half/Period_T;
     writeln(f11,'t_half = ',t_half:7:3,' sec');
     writeln(f11,'N_half = ',N_half:7:3,' cycles');
     writeln('t_half = ',t_half:7:3,' sec');
     writeln('N_half = ',N_half:7:3,' cycles');
    end
   else
    begin
     t_double:=(0.69*t_star)/abs(R11);
     N_double:=t_double/Period_T;
     writeln(f11,'t_double = ',t_double:7:3,' sec');
     writeln(f11,'N_double = ',N_double:7:3,' sec');
     writeln('t_double = ',t_double:7:3,' sec');
     writeln('N_double = ',N_double:7:3,' sec');
    end;
  end
 else
  begin
   Real_Roots(R11);
   Real_Roots(R22);
  end;
end; { Find_Frequency }
begin { Roots }
 New_ChEqn(A1,B1,C1,D1,E1);
```

```
 Ferrari(A1,B1,C1,D1,E1,R1,R2,R3,R4,I1,I2,I3,I4);
 writeln(f11,'_________________________________________');
 writeln(f11);
 writeln(f11,'The roots of the ch eqn are :');
 writeln('The roots of the ch eqn are :');
 writeln(f11,'Lambda1 = ',R1,' + i (',I1,')');
 writeln(f11,'Lambda2 = ',R2,' + i (',I2,')');
 writeln('Lambda1 = ',R1,' + i (',I1,')');
 writeln('Lambda2 = ',R2,' + i (',I2,')');
 writeln(f11,'For the above pair of roots :');
 writeln('For the above pair of roots :');
 Find_Frequency(R1,R2,I1,I2);
 writeln(f11,'Lambda3 = ',R3,' + i (',I3,')');
 writeln(f11,'Lambda4 = ',R4,' + i (',I4,')');
 writeln('Lambda3 = ',R3,' + i (',I3,')');
 writeln('Lambda4 = ',R4,' + i (',I4,')');
 writeln(f11,'For the above pair of roots :');
 writeln('For the above pair of roots :');
 Find_Frequency(R3,R4,I3,I4);
 writeln(f11);
end;  { Roots }

procedure Read_Gust_Data;
begin
 writeln(output,'Input data for step gust');
 writeln(output,'Max Gust Vel Wmax = ');
 readln(input,Wmax);
 Wg:=Wmax/PlaneProg[Ve];
 FWg:=0; {Derivative of Gust Velocity }
end;

procedure Sin_gust;
begin
 writeln(output,'Input data for sinusoidal gust');
 writeln(output,'Max Amplitude (Gust Vel) Wmax = ');
```

```
  writeln(output,'Input frequency(rad/s) of sinusoidal gust');
  readln(input,Frequency);
  Wg:=(Wmax/PlaneProg[Ve])*sin(Frequency*time);
 end;

procedure RK( Y:matrix;
            var F:matrix);
 begin { RK }
  F[w]:=(-(Ag*Y[u]+Ai*Y[w]+Aj*Y[q]+Ak*Y[theta])+Gv*FWg+Gw*Wg+
   E2*elevator_deflection)/Ah;
  F[u]:=(-(Ab*Y[u]+Ac*F[w]+Ad*Y[w]+Ae*Y[q]+Af*Y[theta])+Gt*FWg+
   Gu*Wg+E1*elevator_deflection)/Aa;
  F[q]:=(-(Al*Y[u]+Am*F[w]+Ao*Y[w]+Aq*Y[q]+Ar*Y[theta])+Gx*FWg+
   Gy*Wg+E3*elevator_deflection)/Ap;
  F[theta]:=Y[q];
 end; { RK }

procedure RKM(Delta_t,elevator_deflection:real;var Y:matrix);
 var
    I:perturbation_quantity;
    Y1,K1,K2,K3,K4:matrix;
    Temporary:matrix;
    Dw              :real;
 begin { of RKM }
  for I:=u to theta do Y1[I]:=Y[I];
  RK(Y1,K1);
  for I:=u to theta do
   begin
    K1[I]:=delta_t*K1[I];
    Y1[I]:=Y1[I]+(K1[I]/2);
   end;
  RK(Y1,K2);
  for I:=u to theta do
   begin
    K2[I]:=delta_t*K2[I];
    Y1[I]:=Y[I];
```

```
    Y1[I]:=Y1[I]+(K2[I]/2);
   end;
  RK(Y1,K3);
  for I:=u to theta do
   begin
    K3[I]:=delta_t*K3[I];
    Y1[I]:=Y[I];
    Y1[I]:=Y1[I]+K3[I];
   end;
  RK(Y1,K4);
  for I:=u to theta do
   begin
    K4[I]:=delta_t*K4[I];
    Y[I]:=Y[I]+(1/6)*(K1[I]+2*K2[I]+2*K3[I]+K4[I]);
   end;
  RK(Y,Temporary);
  Dw:=Temporary[w];
  Load_factor:=-(sqr(PlaneProg[Ve])*(Dw-Y[q]))/(Ch_length*gr);
  writeln(f11,time,' ',Y[u],' ',Y[w], ' ',Y[q],' ',Y[theta],' ',Load_Factor);
  writeln(time,' ',Y[u],' ',Y[w], ' ',Y[q],' ',Y[theta],' ',Load_Factor);
 end; { of RKM }

procedure Print_start(Y:matrix);
 begin
  writeln(f11,time,' ',Y[u],' ',Y[w], ' ',Y[q],' ',Y[theta],' ',Load_Factor);
  writeln(time,' ',Y[u],' ',Y[w], ' ',Y[q],' ',Y[theta],' ',Load_Factor);
 end;

procedure Print_Reqd_Data;
 begin { Print_Reqd_Data }
  writeln(f11);
  writeln(f11,' ':10,'RESPONSE TO STEP GUSTS');
  writeln(f11,' ':10,'______________________');
  writeln(f11);
  writeln(f11);
  writeln(f11,'An aircraft flying at an altitude of ',Initial_altitude:6:1,' ft ');
```

```
  writeln(fi1,'at a speed of ',PlaneProg[Ve]:4:1,' ft/sec encounters a gust as be
  writeln(fi1);
  writeln(fi1,' Step Gust (ft/sec):  W = ',Wmax:4:2,' = constant');
  writeln(fi1);
  writeln(fi1);
  writeln(fi1,' ':2,'Time(sec)',' ':6,'u_hat',' ':8,'w_hat',' ':8,
    'q_hat',' ':8,'theta',' ':8,'Ld_fctr');
  writeln(' ':2,'Time(sec)',' ':6,'u_hat',' ':8,'w_hat',' ':8,
    'q_hat',' ':8,'theta',' ':8,'Ld_fctr');
 end; { Print_Reqd_Data }

procedure Initial_conditions;
 var
  i:perturbation_quantity;
 begin
  for I:=u to theta do Response[I]:=0.0;
 end;

function power( x,y:real ) : real;
 begin { power }
  power := exp(y*ln(x));
 end; { power }

procedure Read_the_Gust;
 begin
  writeln('Input the value of L');
  readln(input,Scale_turbulence);
  writeln('Input the value of Sigma_w');
  readln(input,Sigma_w);
 end;

procedure Von_Karman;
 var
 A,B,C,D :real;
 begin { Von_Karman }
  A := 1.339*Scale_turbulence*big_omega;
```

```
  B := 1+((8*Sqr(A))/3);
  C := Power((1+Sqr(A)),(11/6));
  D := (Sqr(Sigma_w)*Scale_turbulence)/pi;
  psd_w := (D*B)/(C);
 end; { Von_Karman }

procedure Et_Gust_Model;
 var
 A,B,C :real;
 begin { Et_Gust_Model }
  A:=1+3*sqr(big_omega*Scale_turbulence);
  B:=sqr(1+sqr(big_omega*Scale_turbulence));
  C := (Sqr(Sigma_w)*Scale_turbulence)/(2*pi);
  psd_w:= (C*A)/B;
 end; { Et_Gust_Model }

procedure New_TFs;
 var
  Den_real,Den_imag          :real;
  Num_1_real,Num_1_imag      :real;
  Num_2_real,Num_2_imag      :real;
  Num_3_real,Num_3_imag      :real;
  Num_LF_real,Num_LF_imag    :real;
  LF_coeff                   :real;
  Num_4_real,Num_4_imag      :real;
  Num_5_real,Num_5_imag      :real;
  Num_6_real,Num_6_imag      :real;
  Num_7_real,Num_7_imag      :real;
  Num_8_real,Num_8_imag      :real;
  Num_LFe_real               :real;
  Num_LFe_imag               :real;
  Common_Den                 :real;
  Num_u_gust,Num_w_gust      :real;
  Num_q_gust,Num_theta_gust  :real;
  Num_LF_gust                :real;
  Num_u_elev,Num_w_elev      :real;
```

```
  Num_q_elev,Num_theta_elev :real;
  Num_LF_elev               :real;
  Factor                    :real;
 begin { New_TFs }
          { Common denominator for the transfer functions }
  Den_real:=(Power(Frequency,4)*(Aa*Ap*Ah))+
    (-Power(Frequency,2)*(Ab*(Ah*Aq+Ap*Ai-Am*Aj)+Aa*(Ah*Ar+Ai*
    Aq-Am*Ak-Ao*Aj)-Ag*(Ac*Aq+Ad*Ap-Am*Ae)+Al*(Ac*Aj-Ah*Ae)))
      +(Ab*(Ai*Ar-Ao*Ak)-Ag*(Ad*Ar-Ao*Af)+Al*(Ad*Ak-Ai*Af));
  Den_imag:=(-Power(Frequency,3)*(Ab*Ap*Ah+Aa*(Ah*Aq+Ap*Ai-Am*Aj)-
      Ag*Ac*Ap))+
     (Frequency*(Ab*(Ah*Ar+Ai*Aq-Am*Ak-Ao*Aj)+Aa*(Ai*Ar-Ao*Ak)-
      Ag*(Ac*Ar+Ad*Aq-Am*Af-Ao*Ae)+Al*(Ac*Ak+Ad*Aj-Ah*Af-
      Ae*Ai)));
       { The Gust transfer functions }
  Num_1_real:= (power(Frequency,4)*(-Gv*Ap*Ac+Gt*Ap*Ah))+
       (-power(Frequency,2)*(Gu*(Ap*Ai+Ah*Aq-Am*Aj)-Gw*(Ac*Aq+
Ad*Ap-Am*Ae)-Gv*(Ac*Ar+Ad*Aq-Am*Af-Ao*Ae)+Gy*(Ac*Aj-
Ah*Ae)+Gx*(Ac*Ak+Ad*Aj-Ah*Af-Ae*Ai)+Gt*(Ar*Ah+Ai*Aq-
Am*Ak-Ao*Aj)))+
(Gu*(Ai*Ar-Ao*Ak)-Gw*(Ad*Ar-Ao*Af)+Gy*(Ad*Ak-Ai*Af));
  Num_1_imag:=(-power(Frequency,3)*(Gu*Ap*Ah-Gw*Ap*Ac-Gv*(Ac*Aq+Ad*Ap-
       Am*Ae)+Gx*(Ac*Aj-Ah*Ae)+Gt*(Ap*Ai+Ah*Aq-Am*Aj)))+
       (Frequency*(Gu*(Ar*Ah+Ai*Aq-Am*Ak-Ao*Aj)-Gw*(Ac*Ar+Ad*Aq
-Am*Af-Ao*Ae)-Gv*(Ad*Ar-Ao*Af)+Gy*(Ac*Ak+Ad*Aj-Ah*Af-
Ae*Ai)+Gx*(Ad*Ak-Ai*Af)+Gt*(Ai*Ar-Ao*Ak)));
  Num_2_real:=(power(Frequency,4)*(Aa*Gv*Ap))+
      (-power(Frequency,2)*(Aa*(Gv*Ar+Gw*Aq-Gx*Ak-Gy*Aj)+
       Ab*(Gw*Ap+Gv*Aq-Gx*Aj)-Ag*(Gu*Ap-Gx*Ae+Gt*Aq)-(Gv*Ae-
       Gt*Aj)*Al))+
(Ab*(Gw*Ar-Gy*Ak)-Ag*(Ar*Gu-Gy*Af)+Al*(Gu*Ak-Gw*Af));
  Num_2_imag:=(-power(Frequency,3)*(Ab*Gv*Ap+Aa*(Gw*Ap+Gv*Aq-Gx*Aj)+
       Gt*Ap))+
       (Frequency*(Aa*(Gw*Ar-Gy*Ak)+Ab*(Gv*Ar+Gw*Aq-Gx*Ak-Gy*
       Aj)-Ag*(Gu*Aq-Gx*Af-Gy*Ae+Gt*Ar)+Al*(Gu*Aj-Gv*Af-Gw*Ae
       +Gt*Ak)));
```

```
  {Numerator for Theta}
  Num_3_real:=(-power(Frequency,2)*(Aa*(Ah*Gy+Ai*Gx-Am*Gw-Ao*Gv)+
Ab*(Ah*Gx-Am*Gv)-Ag*(Ac*Gx-Am*Gt)+Al*(Ac*Gv-Ah*Gt)))+
(Ab*(Ai*Gy-Ao*Gw)-Ag*(Ad*Gy-Ao*Gu)+Al*(Ad*Gw-Ai*Gu));
  Num_3_imag:=(-power(Frequency,3)*(Aa*(Ah*Gx-Am*Gv)))+
      (Frequency*(Aa*(Ai*Gy-Ao*Gw)+Ab*(Ah*Gy+Ai*Gx-Am*Gw-
       Ao*Gv)-Ag*(Ac*Gy+Ad*Gx-Am*Gu-Ao*Gt)+Al*(Ac*Gw+Ad*Gv-
       Ah*Gu-Ai*Gt)));
        { The elevator transfer functions}
  Num_4_real:= (-power(Frequency,2)*(E1*(Ai*Ap+Ah*Aq-Am*Aj)-E2*(Ad*Ap+
Ac*Aq-Am*Ae)+E3*(Ac*Aj-Ah*Ae)))+
(E1*(Ai*Ar-Ao*Ak)-E2*(Ad*Ar-Ao*Af)+E3*(Ad*Ak-Ai*Af));
  Num_4_imag:= (-power(Frequency,3)*(Ap*(E1*Ah-E2*Ac)))+
       (Frequency*(E1*(Ar*Ah+Ai*Aq-Am*Ak-Ao*Aj)-E2*(Ad*Aq+
Ac*Ar-Am*Af-Ao*Ae)+E3*(Ac*Ak+Ad*Aj-Ah*Af-Ai*Ae)));
  Num_5_real:= (-power(Frequency,2)*(-E1*Ap*Ag+E2*(Ab*Ap+Aa*Aq)-
E3*Aa*Aj))+
(-E1*(Ag*Ar-Al*Ak)+E2*(Ab*Ar-Al*Af)-E3*(Ab*Ak-Ag*Af));
  Num_5_imag:= (-power(Frequency,3)*(E2*Aa*Ap))+
       (Frequency*(-E1*(Ag*Aq-Al*Aj)+E2*(Aa*Ar+Ab*Aq-Al*Ae)-
E3*(Aa*Ak+Ab*Ak-Ag*Ae)));
      { Numerator for ElevatorTheta }
  Num_6_real:= (-power(Frequency,2)*(E3*Aa*Ah-E2*Aa*Am))+
(E1*(Ao*Ag-Al*Ai)-E2*(Ab*Ao-Al*Ad)+E3*(Ab*Ai-Ag*Ad));
  Num_6_imag:= (Frequency*(E1*(Am*Ag-Al*Ah)-E2*(Aa*Ao+Ab*Am-Ac*Al)+
E3*(Ab*Ah+Aa*Ai-Ac*Ag)));
       { Numerator of TF q (gust) }
  Num_7_real:= -Frequency*Num_3_imag;
  Num_7_imag:=  Frequency*Num_3_real;
               { Numerator of TF q (elev) }
  Num_8_real:= -Frequency*Num_6_imag;
  Num_8_imag:=  Frequency*Num_6_real;
  LF_coeff :=  -(2*sqr(PlaneProg[Ve]))/(gr*PlaneProg[c]);
  Num_LF_real:= LF_coeff*((-Frequency*Num_2_imag)-Num_7_real);
  Num_LF_imag:= LF_coeff*((Frequency*Num_2_real)-Num_7_imag);
  Num_LFe_real:= (-gr/PlaneProg[CLe])*((-Frequency*Num_5_imag)-Num_8_real);
```

```
  Num_LFe_imag:= (-gr/PlaneProg[CLe])*((Frequency*Num_5_real)-Num_8_imag);
         { Modify Common denominator for Transfer Function theta }
  Common_Den:= sqr(Den_real)+sqr(Den_imag);
  Num_u_gust:= sqr(Num_1_real)+sqr(Num_1_imag);
  Num_w_gust:= sqr(Num_2_real)+sqr(Num_2_imag);
  Num_q_gust:= sqr(Num_7_real)+sqr(Num_7_imag);
  Num_theta_gust:= sqr(Num_3_real)+sqr(Num_3_imag);
  Num_LF_gust:= sqr(Num_LF_real)+sqr(Num_LF_imag);
  Num_u_elev:= sqr(Num_4_real)+sqr(Num_4_imag);
  Num_w_elev:= sqr(Num_5_real)+sqr(Num_5_imag);
  Num_q_elev:= sqr(Num_8_real)+sqr(Num_8_imag);
  Num_theta_elev:= sqr(Num_6_real)+sqr(Num_6_imag);
  Num_LF_elev:= sqr(Num_LFe_real)+sqr(Num_LFe_imag);
  Factor:= 1/PlaneProg[Ve];
  TF_u_gust:= Factor*sqrt(Num_u_gust/Common_Den);
  TF_w_gust:= Factor*sqrt(Num_w_gust/Common_Den);
  TF_q_gust:= Factor*sqrt(Num_q_gust/Common_Den);
  TF_theta_gust:= Factor*sqrt(Num_theta_gust/Common_Den);
  TF_LF_gust:= Factor*sqrt(Num_LF_gust/Common_Den);
  Ph_u_gust:= (180/pi)*(arctan(Num_1_imag/Num_1_real)-
arctan(Den_imag/Den_real));
  Ph_w_gust:= (180/pi)*(arctan(Num_2_imag/Num_2_real)-
arctan(Den_imag/Den_real));
  Ph_q_gust:= (180/pi)*(arctan(Num_7_imag/Num_7_real)-
arctan(Den_imag/Den_real));
  Ph_theta_gust:= (180/pi)*(arctan(Num_3_imag/Num_3_real)-
    arctan(Den_imag/Den_real));
  Ph_LF_gust:= (180/pi)*(arctan(Num_LF_imag/Num_LF_real)-
 arctan(Den_imag/Den_real));
  TF_u_elev:= sqrt(Num_u_elev/Common_Den);
  TF_w_elev:= sqrt(Num_w_elev/Common_Den);
  TF_q_elev:= sqrt(Num_q_elev/Common_Den);
  TF_theta_elev:= sqrt(Num_theta_elev/Common_Den);
  TF_LF_elev:= sqrt(Num_LF_elev/Common_Den);
  Ph_u_elev:= (180/pi)*(arctan(Num_4_imag/Num_4_real)-
arctan(Den_imag/Den_real));
```

```
  Ph_w_elev:= (180/pi)*(arctan(Num_5_imag/Num_5_real)-
arctan(Den_imag/Den_real));
  Ph_q_elev:= (180/pi)*(arctan(Num_8_imag/Num_8_real)-
arctan(Den_imag/Den_real));
  Ph_theta_elev:= (180/pi)*(arctan(Num_6_imag/Num_6_real)-
    arctan(Den_imag/Den_real));
  Ph_LF_elev:= (180/pi)*(arctan(Num_LFe_imag/Num_LFe_real)-
 arctan(Den_imag/Den_real));
  TF_u_gust:= sqr(TF_u_gust);
  TF_w_gust:= sqr(TF_w_gust);
  TF_q_gust:= sqr(TF_q_gust);
  TF_theta_gust:= sqr(TF_theta_gust);
  TF_LF_gust:= sqr(TF_LF_gust);
 end; { New_TFs }

procedure Get_Gain_Factors (var pr:glnp);
 begin { Get_Gain_Factors }
  read(kii,pr[1]);
  read(kii,pr[2]);
  read(kii,pr[3]);
  read(kii,pr[4]);
  read(kii,pr[5]);
  read(kii,pr[6]);
  read(kii,pr[7]);
  readln(kii);
  writeln(fii,'Kalpha_gust = ',pr[1]);
  writeln(fii,'T_Filter = ',pr[2]);
  writeln(fii,'Ku = ',pr[3]);
  writeln(fii,'Kalpha = ',pr[4]);
  writeln(fii,'Kq = ',pr[5]);
  writeln(fii,'Ktheta = ',pr[6]);
  writeln(fii,'KLdFctr = ',pr[7]);
 end; { Get_Gain_Factors }

procedure Gain_Factors(pr:glnp);
 begin
```

```
  Kalpha_gust:=pr[1];
  T_Filter:=pr[2];
  Ku:=pr[3];
  Kalpha:=pr[4];
  Kq:=pr[5];
  Ktheta:=pr[6];
  KLdFctr:=pr[7];
 end;

procedure Out_spectra;
 begin
  Out_u_gust:=TF_u_gust*psd_w;
  Out_w_gust:=TF_w_gust*psd_w;
  Out_q_gust:=TF_q_gust*psd_w;
  Out_theta_gust:=TF_theta_gust*psd_w;
  Out_LF_gust:=TF_LF_gust*psd_w;
 end;

procedure Print_ing;
 begin
  writeln(f11,' ','L = ',Scale_turbulence:3:2,' ':5,'Sigma = ',
   Sigma_w:3:3);
  writeln(f11,' ':2,'big_omega',' ':6,'Frequency',' ':3,'PSD_W',
  ' ':9,'TFu',' ':10,'TFw',' ':10,'TFq',' ':7,'TF_LdFctr',
  ' ':6,'Out_u',' ':8,'Out_w',' ':8,'Out_q',' ':6,
  'Out_LdFctr');
  writeln(f11,' ':3,'(rad/ft)',' ':6,'(Hz)');
  writeln(p11,'THE TRANSFER FUNCTIONS FOR THE ELEVATOR');
  writeln(p11,' ','L = ',Scale_turbulence:3:2,' ':5,'Sigma = ',
   Sigma_w:3:3);
  writeln(p11,' ':2,'big_omega',' ':6,'Frequency',' ':3,'PSD_W',
      ' ':9,'TFu',' ':10,'TFw',' ':10,'TFtheta',' ':3);
  writeln(p11,' ':3,'(rad/ft)',' ':6,'(Hz)');
  writeln(' ','L = ',Scale_turbulence:3:2,' ':5,'Sigma = ',
   Sigma_w:3:3);
  writeln(' ':2,'big_omega',' ':6,'Frequency',' ':3,'PSD_W',
```

```
' ':9,'TFu',' ':10,'TFw',' ':10,'TFq',' ':7,'TF_LdFctr',
' ':6,'Out_u',' ':8,'Out_w',' ':8,'Out_q',' ':6,
'Out_LdFctr');
writeln(' ':3,'(rad/ft)',' ':6,'(Hz)');
end;

procedure Print_out;
begin
 writeln(f11,big_omega:10,' ',dim_frequency:10,' ',psd_w:10,' ',
 TF_u_gust:10,' ',TF_w_gust:10,' ',TF_q_gust:10,' ',
 TF_LF_gust:10,' ',Out_u_gust:10,' ',Out_w_gust:10,' ',
 Out_q_gust:10,' ',Out_LF_gust:10);
 writeln(p11,big_omega:10,' ',dim_frequency:10,' ',psd_w:10,' ',
  TF_u_elev:10,' ',TF_w_elev:10,' ',TF_theta_elev:10);
 writeln(big_omega:10,' ',dim_frequency:10,' ',psd_w:10,' ',
 TF_u_gust:10,' ',TF_w_gust:10,' ',TF_q_gust:10,' ',
 TF_LF_gust:10,' ',Out_u_gust:10,' ',Out_w_gust:10,' ',
 Out_q_gust:10,' ',Out_LF_gust:10);
end;

procedure Graf_Print_out;
begin
 writeln(kk1,big_omega,' ',psd_w);
 writeln(kk2,big_omega,' ',TF_u_gust,' ',TF_w_gust,' ',TF_q_gust,
 ' ',TF_LF_gust);
 writeln(kk3,big_omega,' ',Out_u_gust,' ',Out_w_gust,' ',
 Out_q_gust,' ',Out_LF_gust);
 writeln(kk4,big_omega,' ',Frequency,' ',TF_u_elev,' ',TF_w_elev,
 ' ',TF_theta_elev);
end;

procedure Integrate(ik:integer;Out_quantity:real;
    var sum:real);
begin { Integrate }
 if ((ik = 1) or (ik = 999)) then
  sum:=sum+Out_quantity
```

```
  else
   if ((ik mod 2) = 0) then
    sum:=sum+4*Out_quantity
   else
    sum:=sum+2*Out_quantity;
 end; { Integrate }

procedure Relax_St_Stblty;
 begin { Relax_Static_Stability }
  {writeln('Give the value of KFactor_CLaw');
  readln(KFactor_CLaw);}
  KFactor_CLaw:=0;
  ACMtheta:=KFactor_CLaw*PlaneProg[CMalpha];
  ACMalpha:=(1-KFactor_CLaw)*PlaneProg[CMalpha];
  writeln(fi1);
  writeln(fi1,'___________________________________');
  writeln(fi1,'KFactor_CLaw =',KFactor_CLaw);
 end; {Relax_Static_Stabilty }

procedure Step_gust_calc;
 begin { Step_gust_calculations }
  Read_Gust_Data;
  Print_Reqd_Data;
  Initial_conditions;
  time:=0.0;
  Load_Factor:=0.0;
  Print_start(Response);
  time:=time+delta_time;
  repeat
   RKM((delta_time/t_star),elevator_deflection,response);
   time:=time+delta_time;
  until (time > Final_time);
 end; { Step_gust_calculations }

procedure Check_Print(j1:integer);
 begin { Check_print }
```

```
 if ((j1=1) or (j1=10) or (j1=100)) then
  count:=100000/j1;
 check_coeff:=big_omega*count;
 if (((check_coeff-trunc(check_coeff)) = 0.0) or (big_omega=3e-4)
      or (big_omega=6e-4) or (big_omega=7e-4)) then
  begin
   Print_out;
   Graf_Print_out;
  end;
end; { Check_Print }

function func(pr:glnp):real;
var
 jj:integer;
begin { func }
 Gain_Factors(pr);
 Dummy_Coeffs;
 sum2:=0;
 delta_omega:=PlaneProg[Ve]/100000;
  for jj := 1 to 999 do
   begin
    big_omega := jj/100000;
    dim_frequency:=big_omega*PlaneProg[Ve];
    Von_Karman;
    {Et_Gust_Model;}
    Frequency:=big_omega*Ch_length;
    New_TFs;
    Out_Spectra;
    Integrate(jj,Out_LF_gust,sum2);
   end;
 comfort_criterion:=sqrt((delta_omega*sum2)/(3*pi));
 func:=comfort_criterion;
end; { func }

PROCEDURE amoeba(VAR p: glmpnp; VAR y: glmp; ndim: integer;
      ftol: real; VAR iter: integer);
```

```
(* Programs using routine AMOEBA must supply an external function
func(pr:glnp):real whose minimum is to be found. They must
also define types
TYPE
   glmpnp = ARRAY [1..mp,1..np] OF real;
   glmp = ARRAY [1..mp] OF real;
   glnp = ARRAY [1..np] OF real;
where mp and np are physical dimensions *)
LABEL 990;
CONST
   alpha=1.0;
   beta=0.5;
   gamma=2.0;
   itmax=500;
VAR
   mpts,j,inhi,ilo,ihi,i: integer;
   yprr,ypr,rtol: real;
   pr,prr,pbar: glnp;
BEGIN { Amoeba }
 writeln('Entering AMOEBA');
 mpts := ndim+1;
 iter := 0;
 WHILE true DO
  BEGIN
   ilo := 1;
   IF (y[1] > y[2]) THEN
    BEGIN
     ihi := 1;
     inhi := 2
    END
   ELSE
    BEGIN
     ihi := 2;
     inhi := 1
    END;
   FOR i := 1 to mpts DO
```

```
 BEGIN
  IF (y[i] < y[ilo]) THEN  ilo := i;
  IF (y[i] > y[ihi]) THEN
   BEGIN
    inhi := ihi;
    ihi := i
   END
  ELSE
   IF  (y[i] > y[inhi]) THEN
   IF (i <> ihi) THEN  inhi := i
 END;
rtol := 2.0*abs(y[ihi]-y[ilo])/(abs(y[ihi])+abs(y[ilo]));
IF (rtol < ftol) THEN GOTO 990;
IF (iter = itmax) THEN
 BEGIN
  writeln('pause in AMOEBA - too many iterations');
  goto 990;
 END;
iter := iter+1;
FOR j := 1 to ndim DO pbar[j] := 0.0;
FOR i := 1 to mpts DO
IF (i <> ihi) THEN
 FOR j := 1 to ndim DO
  pbar[j] := pbar[j]+p[i,j];
FOR j := 1 to ndim DO
 BEGIN
  pbar[j] := pbar[j]/ndim;
  pr[j] := (1.0+alpha)*pbar[j]-alpha*p[ihi,j]
 END;
ypr := func(pr);
IF (ypr <= y[ilo]) THEN
 BEGIN
  FOR j := 1 to ndim DO
   prr[j] := gamma*pr[j]+(1.0-gamma)*pbar[j];
  yprr := func(prr);
  IF (yprr < y[ilo]) THEN
```

```
    BEGIN
     FOR j := 1 to ndim DO p[ihi,j] := prr[j];
     y[ihi] := yprr
    END
   ELSE
    BEGIN
     FOR j := 1 to ndim DO p[ihi,j] := pr[j];
     y[ihi] := ypr
    END
  END
 ELSE
  IF  (ypr >= y[inhi]) THEN
   BEGIN
    IF (ypr < y[ihi]) THEN
     BEGIN
      FOR j := 1 to ndim DO p[ihi,j] := pr[j];
      y[ihi] := ypr
     END;
    FOR j := 1 to ndim DO prr[j] := beta*p[ihi,j]+(1.0-beta)*pbar[j];
    yprr := func(prr);
    IF (yprr < y[ihi]) THEN
     BEGIN
      FOR j := 1 to ndim DO p[ihi,j] := prr[j];
      y[ihi] := yprr
     END
    ELSE
     BEGIN
      FOR i := 1 to mpts DO
       IF (i <> ilo) THEN
BEGIN
         FOR j := 1 to ndim DO
  BEGIN
          pr[j] := 0.5*(p[i,j]+p[ilo,j]);
          p[i,j] := pr[j]
         END;
        y[i] := func(pr)
```

```
          END
        END
      END
     ELSE
      BEGIN
       FOR j := 1 to ndim DO p[ihi,j] := pr[j];
       y[ihi] := ypr
      END
  END;
990:END; { Amoeba }

FUNCTION ran1(VAR idum,glix1,glix2,glix3:integer;
      var glr:galr): real;
(* Programs using RAN1 must declare the following variables
VAR
   glix1,glix2,glix3: integer;
   glr: ARRAY [1..97] OF real;
in the main program. *)
CONST
   m1=259200;
   ia1=7141;
   ic1=54773;
   rm1=3.8580247e-6;   (* 1.0/m1 *)
   m2=134456;
   ia2=8121;
   ic2=28411;
   rm2=7.4373773e-6;   (* 1.0/m2 *)
   m3=243000;
   ia3=4561;
   ic3=51349;
VAR
   jjm: integer;
BEGIN
 IF  (idum < 0)  THEN
  BEGIN
   glix1 := (ic1-idum) MOD m1;
```

```
    glix1 := (ia1*glix1+ic1) MOD m1;
    glix2 := glix1 MOD m2;
    glix1 := (ia1*glix1+ic1) MOD m1;
    glix3 := glix1 MOD m3;
    FOR jjm := 1 to 97 DO
     BEGIN
      glix1 := (ia1*glix1+ic1) MOD m1;
      glix2 := (ia2*glix2+ic2) MOD m2;
      glr[jjm] := (glix1+glix2*rm2)*rm1
     END;
    idum := 1
   END;
   repeat
    glix1 := (ia1*glix1+ic1) MOD m1;
    glix2 := (ia2*glix2+ic2) MOD m2;
    glix3 := (ia3*glix3+ic3) MOD m3;
    jjm := 1 + (97*glix3) DIV m3;
   until ((jjm < 97) AND (jjm > 1));
  ran1 := glr[jjm];
  glr[jjm] := (glix1+glix2*rm2)*rm1;
END;{ of RAN1 }

PROCEDURE memcof(data: glnarray; n,m: integer; VAR pm: real;
      VAR cof: glmarray; wk1,wk2: glnarray; wkm: glmarray);
(* Programs using routine MEMCOF must define the data types
TYPE
   glnarray = ARRAY [1..n] OF real;
   glmarray = ARRAY [1..m] OF real;
and must provide workspace arrays wk1,wk2,wkm with the dimensions
shown in the arguments above. *)
LABEL 991;
VAR
   k,j,i: integer;
   pneum,p,denom: real;
BEGIN
 p := 0.0;
```

```
FOR j := 1 to n DO
 BEGIN
  p := p+sqr(data[j])
 END;
pm := p/n;
wk1[1] := data[1];
wk2[n-1] := data[n];
FOR j := 2 to n-1 DO
 BEGIN
  wk1[j] := data[j];
  wk2[j-1] := data[j]
 END;
FOR k := 1 to m DO
 BEGIN
  pneum := 0.0;
  denom := 0.0;
  FOR j := 1 to n-k DO
   BEGIN
    pneum := pneum+wk1[j]*wk2[j];
    denom := denom+sqr(wk1[j])+sqr(wk2[j])
   END;
  cof[k] := 2.0*pneum/denom;
  pm := pm*(1.0-sqr(cof[k]));
  IF (k <> 1) THEN
   BEGIN
    FOR i := 1 to k-1 DO
     BEGIN
      cof[i] := wkm[i]-cof[k]*wkm[k-i]
     END
   END;
  IF (k = m) THEN GOTO 991;
  FOR i := 1 to k DO
   BEGIN
    wkm[i] := cof[i]
   END;
  FOR j := 1 to n-k-1 DO
```

```
    BEGIN
     wk1[j] := wk1[j]-wkm[k]*wk2[j];
     wk2[j] := wk2[j+1]-wkm[k]*wk1[j+1]
    END
  END;
991:END; { of MEMCOF }

FUNCTION evlmem(fdt: real; cof: glmarray; m: integer; pm: real): real;
(* Programs using routine EVLMEM must define the types
TYPE
   glmarray = ARRAY [1..m] OF real;
where m is the dimension of the array of coefficients. *)
VAR
   wr,wi,wpr,wpi,wtemp,theta: double;
   sumi,sumr: real;
   i: integer;
BEGIN
   theta := 6.28318530717959*fdt;
   wpr := cos(theta);
   wpi := sin(theta);
   wr := 1.0;
   wi := 0.0;
   sumr := 1.0;
   sumi := 0.0;
   FOR i := 1 to m DO BEGIN
      wtemp := wr;
      wr := wr*wpr-wi*wpi;
      wi := wi*wpr+wtemp*wpi;
      sumr := sumr-cof[i]*sngl(wr);
      sumi := sumi-cof[i]*sngl(wi)
   END;
   evlmem := pm/(sqr(sumr)+sqr(sumi))
END; { of EVLMEM }

procedure Integrate_psd_w;
 var
```

```
 kkj:integer;
begin { of Integrate_psd_w }
 sum1:=0;
 delta_omega:=PlaneProg[Ve]/100000;
 for kkj:=1 to 999 do
  begin
   big_omega:=kkj/100000;
   Von_Karman;
   {Et_Gust_Model;}
   Integrate(kkj,psd_w,sum1);
  end;
 sum1:=sum1*delta_omega;
end; { of Integrate_psd_w }

procedure Rand_Phi;
 begin { of Random_Phi }
  random_Phi:=2*pi*RAN1(Iphi,gplix1,gplix2,gplix3,gphi);
 end; { of Random_Phi }

procedure Generate_random(var random_omega:real; var accept:boolean);
 var
  Compare_func,Total_area_A          :real;
  random_Aprime,random_y             :real;
 begin { Generate_random }
  Compare_func:=3.54336E4/sum1;
  Total_area_A:=Compare_func*(Final_omega-Initial_omega);
  random_Aprime:=Total_area_A*RAN1(Iomega,golix1,golix2,golix3,gomega);
  random_omega:=(random_Aprime/Compare_func)+Initial_omega;
  random_y:=RAN1(Irandom_y,gylix1,gylix2,gylix3,grandom_y);
  big_omega:=random_omega;
  Von_Karman;
  psd_w:=psd_w/sum1;
  if ((psd_w/Compare_func) <= random_y) then
   accept:=true;
  random_omega:=random_omega*PlaneProg[Ve];
 end; { Generate_random }
```

```
var
 summ  :real;
 kkm   :integer;
begin { Rand_Time_Func }
 summ:=0;
 for kkm:=1 to Total_N do
  summ:=summ+cos(Omegak[kkm]*time+Phik[kkm]);
 random_Time_Func:=sqrt(sum1*2/Total_N)*summ;
end; { Rand_Time_Func }

PROCEDURE four1(VAR data: gldarray; nn,isign: integer);
(* Programs using routine FOUR1 must define type
TYPE
   gldarray = ARRAY [1..nn2] OF real;
in the calling routine, where nn2=nn+nn. *)
VAR
   ii,jj,n,mmax,m,j,istep,i: integer;
   wtemp,wr,wpr,wpi,wi,theta: double;
   tempr,tempi: real;
BEGIN
   n := 2*nn;
   j := 1;
   FOR ii := 1 to nn DO BEGIN
      i := 2*ii-1;
      IF (j > i) THEN BEGIN
         tempr := data[j];
         tempi := data[j+1];
         data[j] := data[i],
         data[j+1] := data[i+1];
         data[i] := tempr;
         data[i+1] := tempi
      END;
      m := n DIV 2;
      WHILE ((m >= 2) AND (j > m))  DO BEGIN
         j := j-m;
         m := m DIV 2
```

```
         END;
         j := j+m
      END;
      mmax := 2;
      WHILE (n > mmax) DO BEGIN
         istep := 2*mmax;
         theta := 6.28318530717959/(isign*mmax);
         wpr := -2.0*sqr(sin(0.5*theta));
         wpi := sin(theta);
         wr := 1.0;
         wi := 0.0;
         FOR ii := 1 to (mmax DIV 2) DO BEGIN
            m := 2*ii-1;
            FOR jj := 0 to ((n-m) DIV istep) DO BEGIN
               i := m + jj*istep;
               j := i+mmax;
               tempr := sngl(wr)*data[j]-sngl(wi)*data[j+1];
               tempi := sngl(wr)*data[j+1]+sngl(wi)*data[j];
               data[j] := data[i]-tempr;
               data[j+1] := data[i+1]-tempi;
               data[i] := data[i]+tempr;
               data[i+1] := data[i+1]+tempi
            END;
            wtemp := wr;
            wr := wr*wpr-wi*wpi+wr;
            wi := wi*wpr+wtemp*wpi+wi
         END;
         mmax := istep
      END
   END;

   begin { main }
    Reset(DskF1,'AirplaneData');
    {Reset(DskF1,'AirplaneEtkin');}
    Reset(inp,'vertex.in');
    Reset(k11,'gain.in');
```

```
Rewrite(f11,'tfg.o');
Rewrite(p11,'tfe.o');
Rewrite(kk1,'gpsdw.dat');
Rewrite(kk2,'gtfg.dat');
Rewrite(kk3,'goutg.dat');
Rewrite(kk4,'gtfe.dat');
Rewrite(kk5,'gtime.dat');
ReadDskForACDer;
Am_ConstCoeff;
Read_the_Gust; { for Von Karman model of gust }

{__________ driver for OPTIMISATION _________________}
{ writeln('Reading the vertices of the initial simplex');
 for i:= 1 to mp do
  begin
   for j:=1 to np do
    begin
     read(inp,p[i,j]);
    end;
   readln(inp);
  end;
 ndim := np;
 FOR i := 1 to mp DO
  BEGIN
   FOR j := 1 to np DO x[j] := p[i,j];
   y[i] := func(x);
  END;
 amoeba(p,y,ndim,ftol,iter);
 writeln;
 writeln('Iterations: ',iter:3);
 writeln('Vertices of final simplex and');
 writeln('function values at the vertices:');
 writeln;
 writeln('i':3,'Kalphagust':13,'T_Filter':10,'Ku':12,
 'Kalpha':12,'Kq':12,'Ktheta':12,'KLdFctr':12,
 'function':14);
```

```
 writeln;
 FOR i := 1 to mp DO
  BEGIN
   write(i:3);
   FOR j := 1 to np DO
    BEGIN
     write(p[i,j]:12:6);
    END;
   writeln(y[i]:12:6);
  END;
 writeln;}

{______ driver for CALCULATIONS USING CONTROL LAW -------}
 {Relax_St_Stblty;
 Get_Gain_Factors(pr);
 Gain_Factors(pr);
 Dummy_Coeffs;
 Roots;
 Step_gust_calc;
 sum2:=0;
 delta_omega:=PlaneProg[Ve]/100000;
 Print_ing;
  for j := 1 to 999 do
   begin
    big_omega := j/100000;
    dim_frequency:=big_omega*PlaneProg[Ve];
    Von_Karman;
    Et_Gust_Model;
    Frequency:=big_omega*Ch_length;
    New_TFs;
    Out_Spectra;
    Check_Print(j);
    Integrate(j,Out_LF_gust,sum2);
   end;
 comfort_criterion:=sqrt((delta_omega*sum2)/(3*pi));
 writeln(f11,'Comfort_Criterion = ',comfort_criterion);
```

```
 writeln('Comfort_Criterion = ',comfort_criterion);}

{______ driver for RANDOM GUST CALCULATIONS -------}
 Iomega:=-123;
 Iphi:=-113;
 Irandom_y:=-11;
 accept:=false,
 Integrate_psd_w;
 for nkk1:=1 to Total_N do
  begin
   Rand_Phi;
   Phik[nkk1]:=random_phi;
   repeat
    Generate_random(random_omega,accept);
   until (accept = true),
   Omegak[nkk1]:=random_omega;
  end;
 time:=Initial_time;
 iicount:=0;
 repeat
  Rand_Time_Func;
  iicount:=iicount+1;
  Time_data[iicount]:=random_Time_Func;
  if (iicount < 200) then
   begin
    writeln(kk5,time,Time_data[iicount]);
   end;
  time:=time+delta_time;
 until ( iicount = Total_N );
 for kkm:=1 to Total_N do
  begin
   WK1[kkm]:=0;
   WK2[kkm]:=0;
  end;
 for kkm:=1 to Order_m do WKM[kkm]:=0;
 MEMCOF(Time_data,Total_N,Order_M,PM,COF,WK1,WK2,WKM);
```

```
Relax_St_Stblty;
Get_Gain_Factors(pr);
Gain_Factors(pr);
Dummy_Coeffs;
for kkm:=1 to nn2 do
 begin
  Moutu[kkm]:=0;
  Moutw[kkm]:=0;
  Moutq[kkm]:=0;
  MoutLF[kkm]:=0;
 end;
for kkm:=1 to nnby2 do
 begin
  Frequencys:=kkm/(nn*delta_time);
  big_omega:=Frequencys/Planeprog[Ve];
  dim_frequency:=2*pi*Frequencys;
  fdt:=Frequencys*delta_time;
  psd_w:=EVLMEM(fdt,COF,Order_M,PM);
  if (kkm < nnby2) then
   begin
    writeln(kk1,big_omega,' ',psd_w);
    writeln(big_omega,' ',psd_w);
   end;
  Frequency:=big_omega*Ch_length;
  New_TFs;
  Out_Spectra;
  Moutu[2*kkm+1]:=Out_u_gust;
  Moutw[2*kkm+1]:=Out_w_gust;
  Moutq[2*kkm+1]:=Out_q_gust;
  MoutLF[2*kkm+1]:=Out_LF_gust;
 end;
four1(Moutu,nn,isign);
four1(Moutw,nn,isign);
four1(Moutq,nn,isign);
four1(MoutLF,nn,isign);
time:=Initial_time;
```

```
for kkm:=1 to 199 do
 begin
  writeln(kk3,time,' ',Moutu[2*kkm-1],' ',Moutw[2*kkm-1],
  ' ',Moutq[2*kkm-1],' ',MoutLF[2*kkm-1]);
  writeln(time,' ',Moutu[2*kkm-1],' ',Moutw[2*kkm-1],
  ' ',Moutq[2*kkm-1],' ',MoutLF[2*kkm-1]);
  time:=time+delta_time;
 end;

Close(f11);
Close(inp);
Close(p11);
Close(k11);
Close(kk1);
Close(kk2);
Close(kk3);
Close(kk4);
Close(kk5);
end. { main }
```

Date Slip

This book is to be returned on the date last stamped.